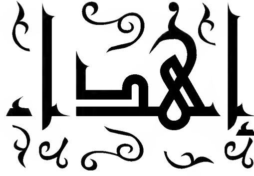
تعرف على
PHP & MySQL

تأليف

صقر بن محمد العنـزي
www.4alarab.com



الى والدي حفظه اللّه الذي علمني ان الرجال وحدهم من يقهرون الصعاب ويصلون الى الهدف .

الى والدتي حفظها اللّه التي علمتني ان المحبة والرحمة هما سر الحياة السعيدة .

الى المرأة التي كانت دائما خلفي تدفعني نحو الإبداع الى زوجتي حفظها اللّه .

الى اطفالي حفظهم اللّه الذين ضحكاتهم تزيدني إصرارا وقوة .

المقدمة

الحمد لله رب العالمين ، والصلاة والسلام على أشرف المرسلين وبعد ،

اخي الكريم بين يديك اول كتاب عربي يتكلم عن البرمجة بلغة php وقواعد البيانات MySQL ، وايضا اول كتاب يتم نشره آليا في المكتبة العربية ، ولنشره آليا القصة التالية :

بعد ان تقصية بعض المعلوات رايت ان نشره عن طريق دور النشر سوف يكون له السلبيات التالية :
- سوف يرتفع سعره .
- سوف يصبح من الصعب الحصول عليه في بعض البلدان العربية .
- سوف يتأخر نشره .

ومع هذه الاسباب ولثقتي في القارئ والمثقف العربي قررت طرحه ككتاب الكتروني وبسعر مناسب دون اية حماية .
وبأذن الله فسوف اطرح بعض الكتب ايضا قريبا في المواضيع التالية :
فيجوال بيسك / اوراكل / قواعد البيانات / الشبكات

لذلك **فقط** اذا اعجبك الكتاب واستفدت منه ولا يشق عليك دفع المبلغ
الرجاء ارسال مبلغ ١٠ ريالات سعودية للحساب التالي :

المملكة العربية السعودية
البنك السعودي الامريكي
حساب رقم
١٠٠١٨٢٢٧١٤
بأسم : صقر بن محمد العنزي
Sager Mohammed alonezi

ونسأل الله ان يهدينا سواء السبيل .

صقر بن محمد العنزي
الرياض - ١٤٢٣ هـ
info@4alarab.com

فهرس المحتويات

تمهيد

لغة الـ Php

تسمية PHP اتت من (Personal Home Page tools) أي ادوات انشاء الصفحات الشخصية وهذه الادوات انتجها في البداية شخص يدعى راسموس ليدروف وترك هذه الادوات مجانية ومفتوحة المصدر ، وهذا جعل الكثيرين يشاركون في تطوير هذه الادوات الى ان اصبحت لغة كبيرة ومهمة جداً خصوصا في عالم الانترنت .

من اهم ما يميز هذا اللغة انها مجانية ومفتوحة المصدر وتستطيع الحصول على الكثير من البرامج المكتوبة بهذه اللغة مجانا من الانترنت بشكل لايصدق .

ماهي الـ MySQL

عندما نتحدث عن البرامج الكبيرة والقوية لابد ان نتحدث عن قواعد البيانات وبالرغم من ان لغة الـ PHP تدعم تقريبا كل محركات قواعد البيانات الا ان مايهمنا هنا هو دعمها لقواعد البيانات MySQL المجانية والمتوفرة في اغلب مواقع الاستضافة المجانية والمدفوعة الاجر ، وبدت الشركات تفضل MySQL على قواعد البيانات المشهورة مثل الاوراكل وغيرها بسبب انها مستقرة ومجانية .

ماذا احتاج لأكتب برنامج بلغة php

تستطيع كتابة برامج php بأستخدام أي محرر نصوص عادي وحفظ الملف بالامتداد php ، وبالنسبة لي فأنا استخدم المفكرة التي تأتي مع وندوز لكتابة برامجي .

كيف اشغل برنامجي

لكي تستطيع تشغيل برنامجك يجب عليك ان تستخدم احد طريقتين :
الاولى : بناء سيرفر شخصي وتجربة برامجك على جهازك .
ثانيا : ارسال الملفات الى موقعك في الانترنت وتجربتها هناك .
وسوف نتحدث عن كلا الطريقتين بالتفصيل .

تركيب سيرفر محلي

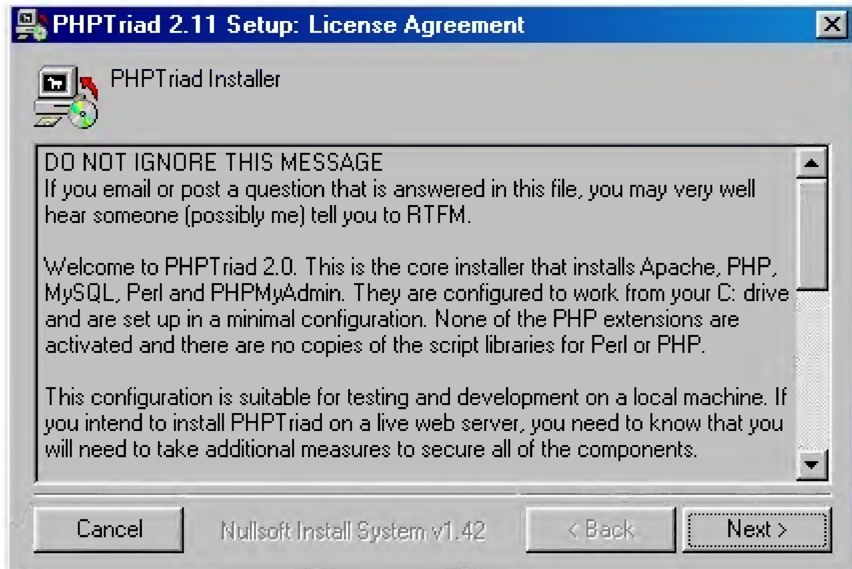
جميع من يكتبون برامج PHP يكون لديهم سيرفر محلي ونقصد بسيرفر محلي هو ان تحول جهازك الشخصي الى موقع استضافة تستطيع من خلالة تجربة برامجك قبل ارسالها الى الانترنت .

عموما لاتقلق هذه العملية سهلة جدا ولن تأخذ اكثر من دقائق وبكل يسر وسهولة واليك الخطوات بالترتيب :

- تحميل برنامج سيرفر في جهازك وانا افضل PHPTriad 2.0 المجاني الذي سوف تجد منه نسخة حديثه في السي دي المرفق بهذا الكتاب في المجلد server اذهب الي المجلد واضغط مرتين على الايقونة .

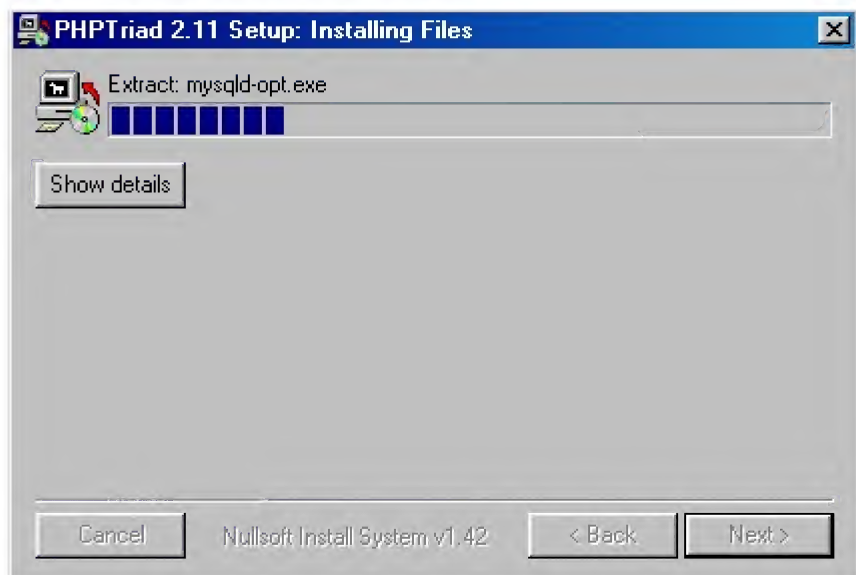


عند تشغيل البرنامج سوف تشاهد الشاشة التالية :



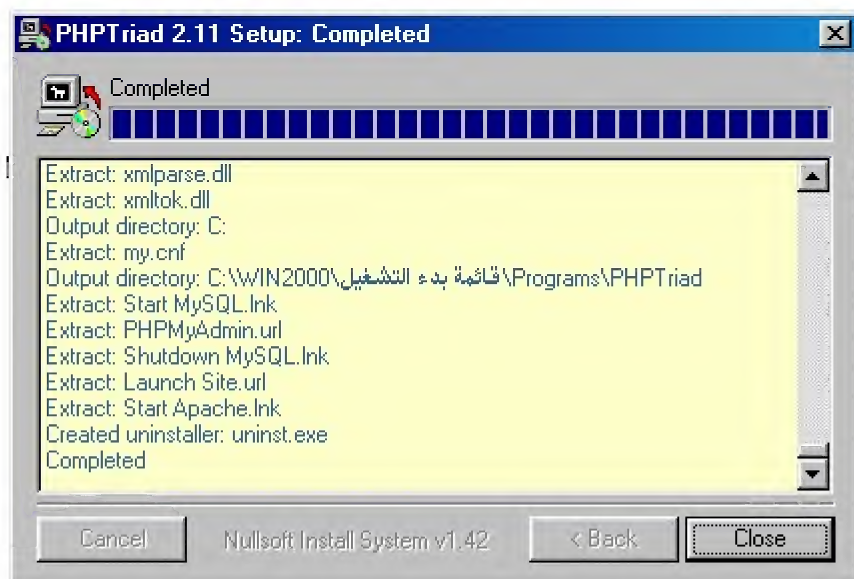
شاشـة تبين شروط استخدام هذا البرنامج ، كل ماعليك فعله الان هو الضغط على الزر Next .

بعد ذلك سوف تشاهد الشاشة التالية :



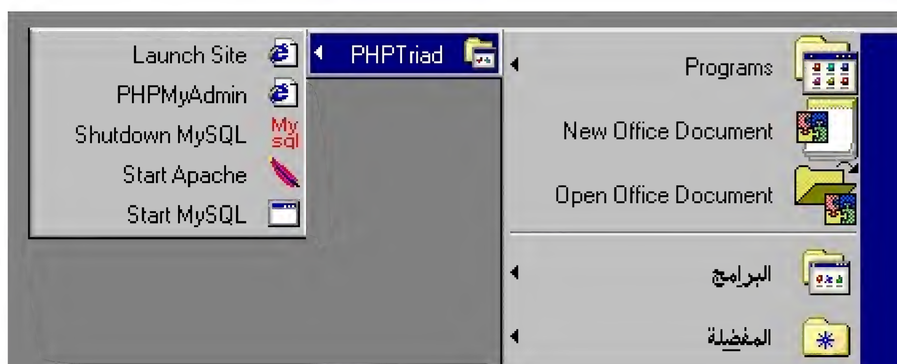
وهـي تـدل علـى انـه الان جـاري نسـخ وتثبيـت الـبرنامج فـي جهازك ، عليك الانـتظار قلـيلا ولو اردت مشاهدة الملفات التي تنسخ الىجهازك اضغط على

الـزر Show details الموجـود فـي الجـزء الاعلـى الايسـر من تلك الشـاشـة ، وبعد نهاية تحميل البرنامج في جهازك سوف تشاهد الشاشة التالية :



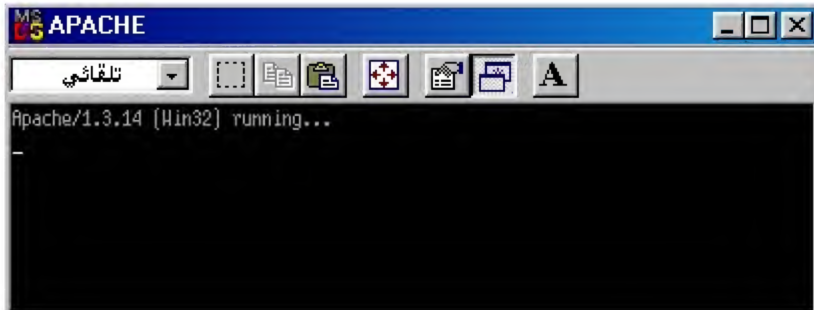
اذن تهانينا لك ، لقد اصبح بأمكانك استخدام جهازك كسيرفر شخصي .

والان لكـي تحـول جهـازك الـى سيرفر شخصي عليك تشغيل البرنامج ، فعند الضـغط علـى الـزر ابـدأ او (Start) وسـوف تجـد فـي اعلـى القائمة مجموعة جديدة اسمها   PHPTriad <=programs انظر الصورة :



ومـن هـذه الشـاشــة نحـن بحاجـة الـى تشغيل خدمتين ليصبح الجهاز سيرفر شخصي متكامل هما :

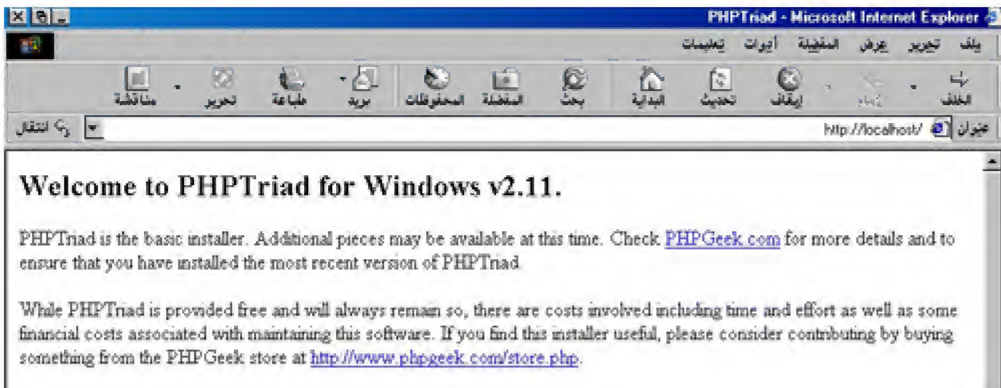
الاولـى : بـرنامج محاكـي السـيرفر الشخصـي الذي يحول الجهاز الى سـيرفر شخصـي عـن طـريقه نسـتطيع تشـغيل جمـيع صـفحات الـوب وبرامج الـ php ونشـغله عـن طـريق الضغط علـى الايقونـة المسـماه Start Apache وعـند الضغط عليها سـوف نشـاهد الشـاشـة التالية :



وهي تعني ان السـيرفر يعمل الان بشـكل سـليم .

الثانـي : وهـو محـرك ومشـغل قواعـد البـيانات MySQL والـذي عـن طـريقه تسـتطيع انشـاء قواعد البيانات وانشـاء الجداول وادخال البيانات فيها واسـتخراج البـيانات مـنها ، ولتشـغيل هـذا المحـرك فقـط اضـغط علـى الايقونة المسماه Start MySQL وسـوف تلاحـظ ظهـور شاشـة دوس سـوداء بشـكل سـريع ثم تختفي ، اذن البرنامج الان يعمل .

الان اصـبح جهازك يعمل كمسـتضيف لصفحات الانترنت فلو شـغلت الاكسـبلورر وكتبـت فـي العـنوان http://localhost سـوف تجـد انـك تصـل الـى الصـفحة الرئيسـية في سـيرفرك المحلي (انظر الصورة) .



وهذا يعني ان السيرفر الشخصي لديك يعمل الان .

لكي ترى ان محرك قواعد البيانات يعمل لديك اذهب الى اسفل الصفحة لترى الرابط PHPMyAdmin انظر الصورة :

Name	Description
PHPMyAdmin	PHPMyAdmin is a web-based interface to MySQL. Use it to manage your databases.

اضغط عليه لترى الصورة التالية :



ومن خلال هذا البرنامج تستطيع تصفح قواعد البيانات المتوفرة لديك ، او انشاء قاعدة بيانات جديدة ، وغيرها من العمليات وسوف نتحدث عن هذا البرنامج وكيفية انشاء قاعدة بيانات به بالتفصيل لاحقا عندما نتحدث عن قواعد البيانات والتعامل معها .

اين نضع الصفحات والبرامج

ان السيرفر الشخصي لا يخدم الجهاز كاملا ، أي ان الصفحات او البرامج الموجودة خارج حدود السيرفر الشخصي لا يمكن التعامل معها كصفحات انترنت نشطه او متفاعله .

اذن اين اضع الصفحات والسكربتات حتى استطيع تصفحها وتنفيذها كصفحة انترنت فاعلة ؟
الجواب بكل سهولة في المجلد (C:\apache\htdocs) .

فبرنامج السيرفر الذي قمنا بتحميله منذ قليل يوجد هذا مجلد ويعتبره العنوان الرئيسي لموقعك . فعندما تكتب localhost في المتصفح ينقلك مباشرة الى هذا المجلد وعند ذلك سوف يحمل الصفحة الرئيسية في هذا المجلد .
اذن دعنا نضرب هذا المثل لمزيد من الشرح فمثلا لو كان لدي ملف اسمه (My.htm) فسوف اضعه في المجلد (C:\apache\htdocs) ولكي اتصفحه علي كتابة مايلي :

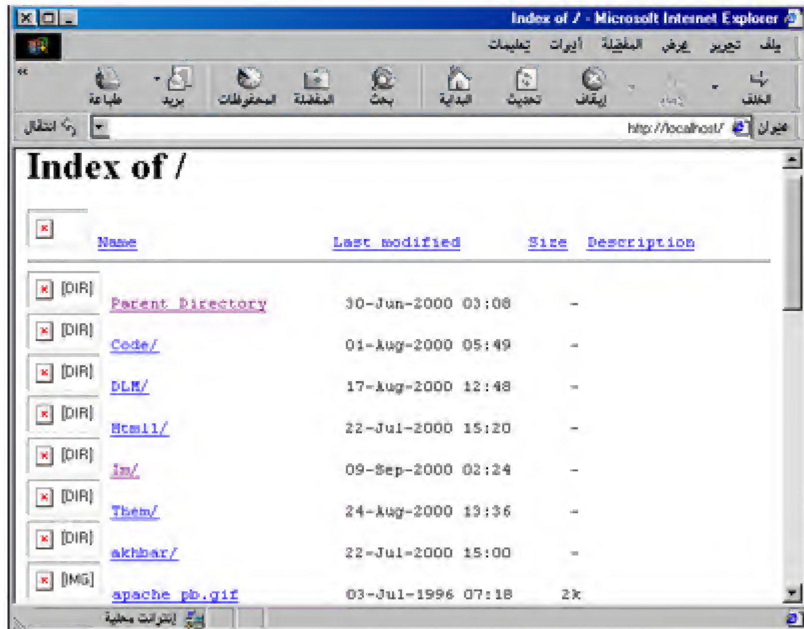
http://localhost/My.htm

وبالطبع يمكنك انشاء مجلدات داخل المجلد الرئيسي وعند استدعاء الملفات الموجودة داخلها يجب الاشارة اليها مثل http://localhost/dir1/My.htm وهكذا .

ماهي الصفحة الرئيسية للمجلد

الصفحة الرئيسية للمجلد عبارة عن ملف يحمل الاسم (index) سواء كان امتداده htm او html او php وعند كتابة عنوان المجلد فقط سوف يتم تحميل هذا الملف مباشرة ، ومن هنا فعند ادخالك عنوان انترنت مثل (www.4alarab.com) فكأنك تكتب (www.4alarab.com/index.htm)
اتمنى ان اكون وضحت الصورة .

وهنا يأتي دور السؤال اذا لم يكن هناك ملف يحمل الاسم (index) في المجلد فماذا سوف يجري ؟
للأجابة على هذا السؤال سوف نقوم بتغيير الاسم الموجود في المجلد (C:\apache\htdocs) من (index.php) الى (indexB.php) ومن ثم سوف نكتب في عنوان المتصفح العبارة localhost لنشاهد الشاشة التالية :



كما تلاحظ فأنه مع عدم وجود هذا الملف فأنه عند طلب عنوان المجلد سوف يتم عرض محتويات المجلد ، اعتقد انك فهمت لماذا يتم وضع هذا الملف في جميع المجلدات وذلك لاعطاء مزيد من السرية على محتويات المجلدات .

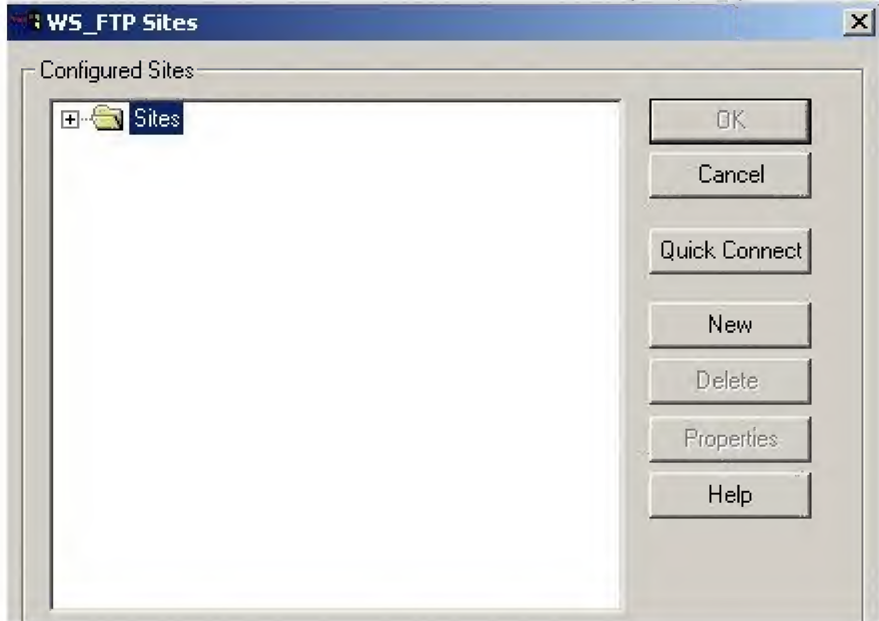
قبل ان نستمر لاتنسى ان تعيد الاسم الى سابق عهده (index.php) .

ارسال البرامج الى الانترنت

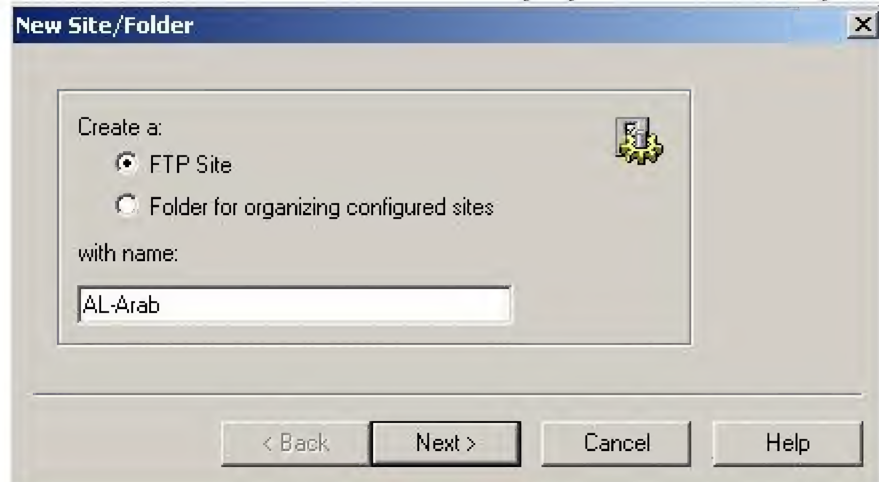
قد تود تجربة برامجك مباشرة في موقعك او ارساله الى موقعك بعد الانتهاء منه وتسأل عن الطريقة المثلى لذلك ، سوف اشرح الان كل مايتعلق بهذا الامر بالتفصيل وبالخطوات المتتابعه :

اولا عليك تركيب برنامج Ftp ويعتبر برنامج WS_FTP من احد البرامج المتميزة في هذا المجال .

عند تشغيل البرنامج تشاهد الشاشة التالية :



وحيث ان هذه المرة الاولى لدخولنا للبرنامج فيجب ان نضغط الزر New للنشأ معلومات اتصال جديد ، وسوف نشاهد الشاشة التالية :



في هذه الشاشة نكتب اسم الاتصال وهو أي اسم نحب ان نستخدمه .
بعد ذلك نضغط على الزر Next لنشاهد الشاشة التالية :

Host Name

Host Name or IP Address:

4alarab.com

< Back | Next > | Cancel | Help

الان اكتب عنوان موقعك او رقم الـ IP له .
بعد ذلك نضغط على الزر Next لنشاهد الشاشة التالية :

Logon Information

User ID:

user

Anonymous

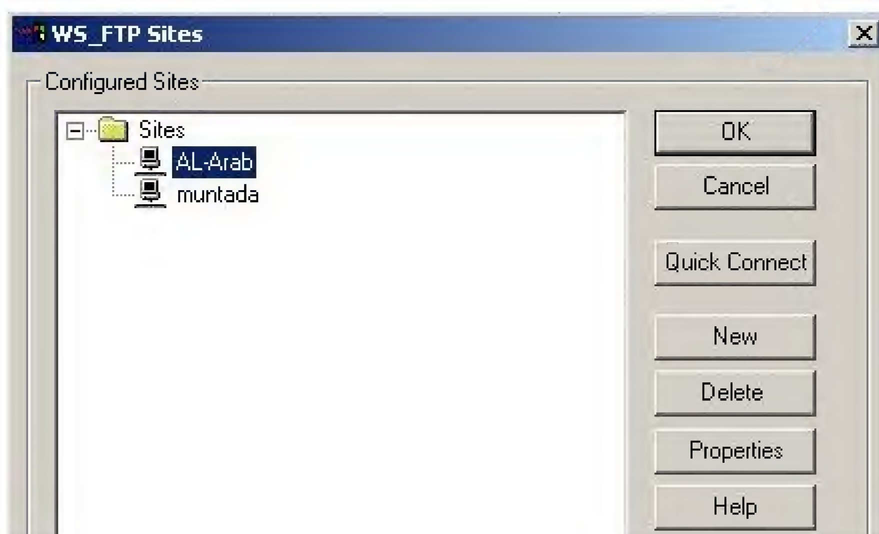
Password:

****

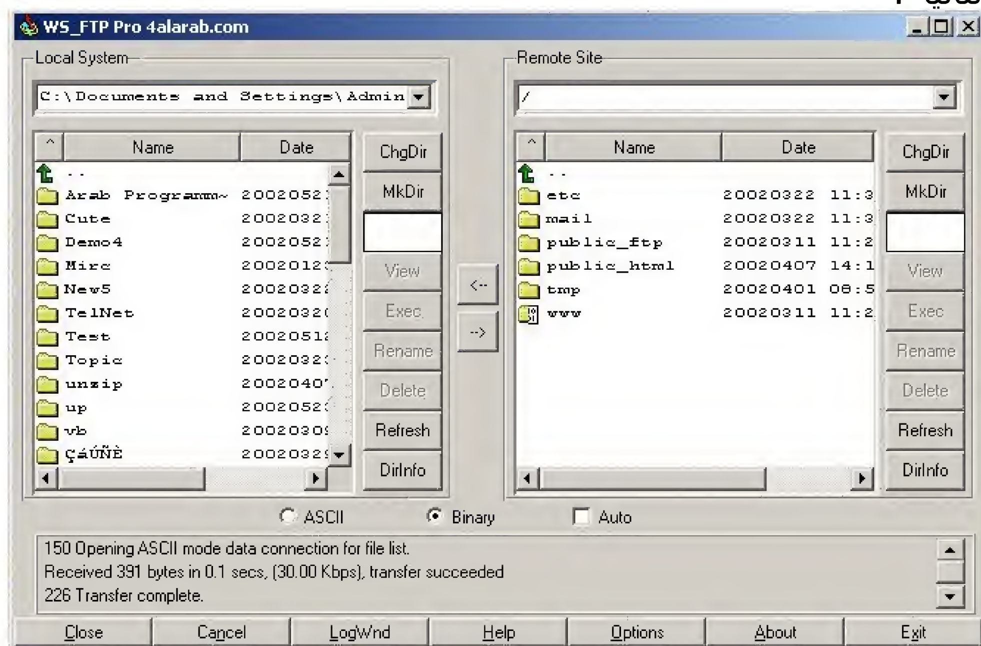
Save Password

< Back | Finish | Cancel | Help

الان اكتب الاسم وكلمة السر المخصصة لدخول موقعك ، هذه المعلومات تحصل عليها من مستضيف موقعك .
الان اضغط على الزر Finish لتشاهد الشاشة التالية :



تشاهد هنا اسماء المواقع الموجودة لديك .
الان اختر الموقع الذي تريد الاتصال به ثم اضغط الزر OK لتشاهد الشاشة التالية :



الان لقد تم الاتصال بنجاح وتشاهد في اليمين موقع على الانترنت وفي اليسار الملفات الموجودة بجهازك .

اول برنامج نكتبه في PHP

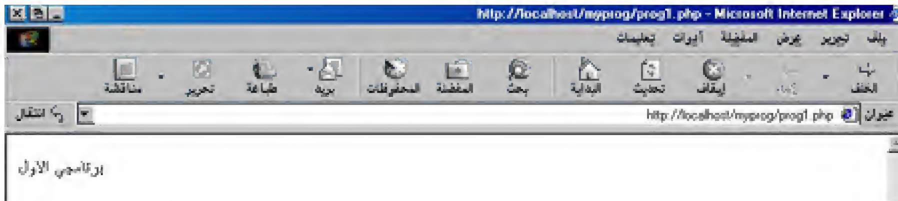
في البداية اعتقد انك قمت بتشغيل السيرفر ومحرك قواعد البيانات .
الان اذهب الى المجلد (C:\apache\htdocs) وانشأ مجلد جديد وليكن اسمه (myprog) .
الان شغل المفكرة واكتب الاوامر التالية :

```
<?php

Echo "برنامجي الاول " ;

?>
```

الان احفظ هذا الملف بالاسم prog1.php في المجلد C:\apache\htdocs\myprog الان اكتب في المتصفح http://localhost/myprog/prog1.php يجب ان يكون لديك في المتصفح الجملة (برنامجي الاول) فقط (انظر الصورة) .



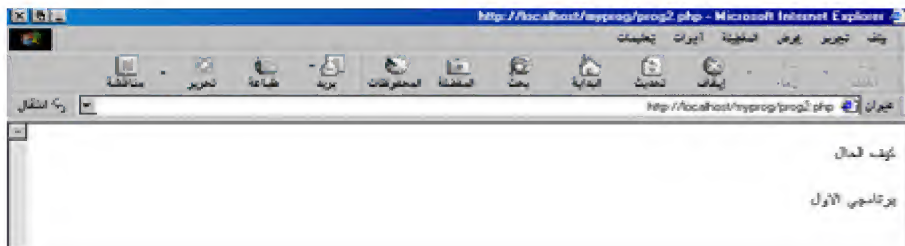
اذن دعنا نتعرف على ماذا فعلنا في هذا المثال البسط ، في البداية اخبرنا السيرفر ان الجمل القادمة هي لغة php فعليه ان يرسلها الى مترجم هذه اللغة وهو الاباتشي ليحولها الى لغة مفهومة للمتصفح وفي السطر الثاني كتبنا امر طباعة للجملة برنامجي الاول وفي السطر الاخير اخبرنا السيرفر نهاية هذه الجمل .

ملاحظة مهمة : لقد ذكرت في الجملة السابقة اننا اخبرنا السيرفر ان هذه الجمل للغة php اذن ماذا يحدث لو لم نخبره ؟
ان هذا السؤال من اهم الاسئلة في تصميم الصفحات وبرمجتها ، فالسيرفر يعتبر أي جمل يقرئها على انها جمل HTML أي صفحات انترنت عادية حتى نخبره بعكس ذلك وهنا تكمن ميزة برمجة الصفحات فلو اردنا مثلا ان نجعل شكل الصفحة من اليمين الى اليسار ونضع بعض اللمسات عليها ثم في الوسط نجعل هناك عداد للزوار فاننا نحتاج الى المزج بين عدة لغات واغلبنا لديه المعرفة الجيدة بلغة النص المتشعب (HTML) واذا لم يكن فيستطيع استخدام احد البرامج مثل الفرونت بيج ونسخ النص ونقله الى برنامجه ، لنأخذ المثال التالي :

في هذا المثال سوف نعدل على مثالنا السابق قليلا :

```
<html dir=rtl >
<p/>  كيف الحال <P>
<?php
echo "برنامجي الاول " ;
?>
</html>
```

الان احفظ هذا الملف بالاسم prog2.php في المجلد C:\apache\htdocs\myprog الان اكتب في المتصفح http://localhost/myprog/prog2.php وسوف تشاهد التالي :



تلاحظ اننا هنا مزجنا بين لغتي النص المتشعب ولغة الـ php فالسطرين الاولين للغة النص المتشعب اما الثالث والرابع والخامس فلغة الـ php واخيرا في الخامس نعود للنص المتشعب .

الطرق المتوفره لنخبر السيرفر عن لغة PHP

كما قلنا سابقا فأن السيرفر سوف يعتبر أي ملف كـ html ولكي نجعل السيرفر يتعرف اننا سنبدأ في ادخال اوامر php فعلينا استخدام احدى الطرق الاربعة التالية :

- الطريقة الاولى :

```
<?
Echo " برنامجي الاول" ؛
?>
```

- الطريقة الثانية :

```
<?php
Echo " برنامجي الاول" ؛
?>
```

- الطريقة الثالثة :

```
<script language="php">
Echo " برنامجي الاول" ؛
</script>
```

- الطريقة الرابعة :

```
<%
Echo " برنامجي الاول" ؛
%>
```

ونستطيع ادخال هذه الطرق في ملف php ففي كل طريقة هناك بداية الكود ونهايته وهنا اعتقد انه اصبح واضح معنا قولنا التمازج فيمكنا كتابة كود html ثم كود php ثم كود html مرة اخرى وكود php كذلك ونستطيع الاستمرار هكذا كما نشاء .

المتغيرات

في أي لغة برمجة تعتبر المتغيرات وطرق كتابتها من اهم الاساليب البرمجية فدائما نشير للمستخدم كمتغير وكلمة السر كمتغير ، لانه بواسطة المتغيرات نستطيع التحكم ببرنامجنا بيسر وسهولة .
من المميزات الرائعة في لغة php انها آليا تعرف نوع المتغير بل وتقوم بالتحويل بين المتغيرات بكل سهولة فيمكن ان تجعل متغير نص ثم تقوم بتحويله الى رقمي بمجرد كتابة قيمة رقمية به .

انواع المتغيرات المدعومة بلغة php

اولا : الاعداد الصحيحيحة (Integers) :
هذا النوع من المتغيرات يأخذ القيم الرقمية الصحيحة الموجبة والسالبة
مع العلم ان php تعتبر أي رقم بهذه الخصائص وغير محاط بعلامة تنصيص عدد صحيح .

امثلة :

```
$a = 1234;
$a = -1234;
```

ثانيا : الاعداد ذات العلامة العشرية (Floating point numbers) :
هذا النوع من المتغيرات يأخذ القيم الرقمية التي تحتوي على علامات عشرية .
مع العلم ان php تعتبر أي رقم بهذه الخصائص وغير محاط بعلامة تنصيص عدد ذو علامة عشرية .
امثلة :

```
$a = 1.234;
$a = 1.2e3;
```

ثالثا : المتغيرات النصية (string) :
المتغير النصي يأخذ حروف وارقام وعلامات مختلفة و تعتبر لغة php أي قيمة بين علامتي تنصيص على انها متغير نصي .
امثلة :

```
$str = "This is a string 123456789>>>> ";
```

رابعا : المصفوفات (Arrays) :
وهي سلسلة من القيم في متغير واحد يعبر عنه ككائن يحتوى على العديد من القيم ويمكن التعبير عن قيم المصفوفة بالارقام او الاسماء حسب الحاجة

، ومن مهام المصفوفات تسهيل البرمجة وتوفير الوقت والجهد فتستطيع قراءة ملف بجملة واحدة لتجد ان كل سطر في الملف وضع داخل المصفوفة وسوف نتعرف على ذلك عندما نتحدث عن كيفية القراءة من الملفات وكذلك قواعد البيانات وغيرها .

امثلة :

```
$a[0] = "abc";
$a[1] = "def";
$b["foo"] = 13;
```

القفز بين انواع المتغيرات

تقوم لغة php بتغيير نوع المتغير اذا غيرت القيمة التي يحتويها فاذا كانت القيمة رقمية فانه سوف يجعله عدد صحيح واذا كتب بعد ذلك في نفس المتغير قيمة نصية فسوف يحوله آليا الى متغير نصي واليك الامثلة :

```
$a=1 ;  // هذا عدد صحيح
$a="sager" ;  //الان نص
$a=1 ;  // عاد رقمي
```

كيفية كتابة المتغيرات :

تعتبر لغة الـ php أي كلمة مسبوقة بعلامة ($) على انها متغير اليك المثال التالي :

```
؟>php
$var1 = "my ";
$var2 = "program";
echo "$var1 $var2";
<؟
```

فلدينا هنا متغيرين في السطر الاول والثاني كل منهما يحمل قيمة وفي السطر الثالث نطلب طباعة هذين المتغيرين لنحصل على الجملة my program .

وهنا يجب ان نأخذ بعين الاعتبار الى ان المتغيرات في الـ PHP لايمكن ان تبدأ برقم ، وايضا ان هناك اختلاف بين الحروف الصغيرة والكبيرة أي s لاتساوي S .

العمليات الحسابية على المتغيرات الرقمية

العملية	الصيغة
جمع متغيرين	$a + $b
طرح متغيرين	$a - $b
ضرب متغيرين	$a * $b
قسمة متغيرين	$a / $b
نسبة المتغير الاول من الثاني	$a % $b
اضافة واحد الى قيمة المتغير	++$a
طرح واحد من قيمة المتغير	--$a

المقارنة بين المتغيرات

العملية	الصيغة
المتغيران متساويان	$a == $b
المتغيران متساويان ومن نفس النوع	$a === $b
المتغيران غير متساويان	$a != $b
المتغيران غير متساويان او ليسا من النوع نفسه	$a !== $b
المتغير الاول اقل من الثاني	$a < $b
المتغير الاول اكبرمن الثاني	$a > $b
المتغير الاول اقل او يساوي الثاني	$a <= $b
المتغير الاول اكبر او يساوي الثاني	$a >= $b

الان لنأخذ بعض الامثلة المهمة على المتغيرات :

مثال ١ :

```
$a = 3;
$a += 5;
```

المتغير كان يحمل القيمة ٣ ثم اضفنا اليه ٥ فالناتج يكون ٨ .

نتـيجة : اذن اذا كـنا نـريد ان نجمـع أي قيمة رقمية على متغير نكتب (=+) ثم القيمة .

مثال ٢ :

```
$b = "my ";
$b .= "program";
```

المتغـير كـان يحمـل النص my ثم طلبنا اضافة النص program ليصبح النص الموجود في المتغير هو my program .
نتـيجة : اذن اذا كـنا نـريد ان نضـيف الـى متغـير نصـي نكتـب (=.) ثـم النص المراد اضافته .

مثال ٣ :

```
$a = “sager” ;
$$a = “My Name” ;
```

وضـعنا فـي المتغير الاول النص sager وفي السـطر الثاني طلبنا من البرنامج ان ينشأ لن متغير اسـمه النص الموجود في المتغير a ويضع فيه القيمة My Name أي اصـبح لديـنا متغيرين هما $a و $sager كما لو اننا كتبنا الجملة التالية :

```
$a = “sager” ;
$sager = “My Name” ;
```

نسـتخدم هـذه الطـريقة عـندما نريد ان نربط متغير معين بمتغيرات اخرى مثل الاسـم وكلمة السـر .

كيف نضيف تعليق

عـندما نكتـب بـرنامج ونعـود الـيه بعد مدة نجد اننا ننسـى الكثير من الخطوات التـي قمـنا بهـا او لمـاذا كتبـنا هـذا الامر او ذاك ، او عمل هذه الدالة او تلك ، بالإضـافة الـى ان هـذا الـبرنامج قـد يقـوم اشـخاص آخرين بالتعديل عليه لذلك تعتـبر التعلـيقات مـن الامـور التـي تبيـن ان المـبرمج محـترف وعملـه منظم ، ولكتابة التعليقات نتبع التالي :

لاجل كتابة تعليق من سطر واحد نستخدم // ثم التعليق الذي نريد
مثال :

```
<?php
// هذه ملاحظه
Echo "line1" ;
// ملاحظة اخرى
Echo "line2" ; // ملاحظة خلف سطر امر مباشره
?>
```

بيـنما لـو اردنـا ان نكتـب تعلـيق مـن عـدة اسطر فاننا نسـتخدم */ في بداية التعليق ونسـتخدم /* لنخبر البرنامج ان ملاحظاتنا انتهت .
مثال :

```
<?php
Echo "my program" ;
/*
السطر الاول من الملاحظات
 السطر الثاني
السطر الثالث من الملاحضات
*/
?>
```

وكمـا تلاحظ اننا نسـتطيع ان نكتب التعليقات في أي مكان من البرنامج ، فكل ماعلينا فعله هو اخبار البرنامج ان هذا السطر او السطر التالي ملاحظة .

المسارات في php

عندما نتحدث عن ملف معين فإننا نذكر انه يجب وضع المسار الكامل له خصوصا اذا لم يكن في نفس المجلد الموجود به البرنامج الذي كتبته ، وعموما هناك عدة حالات لمسار الملف بحيث تتعرف عليه الدوال المختلفة :

لنفترض اننا نود التعامل مع الملف "file.txt" فيكون لدينا احد الحالات التالية :

الحالة الأولى : ان يكون في نفس المجلد الذي ننفذ فيه برنامجنا وهنا نستطيع كتابة اسم الملف فقط دون الحاجة الى إلية إضافات مثل "file.txt" .

الحالة الثانية : ان يكون الملف في مجلد اسمه data داخل المجلد الموجود به برنامجنا فعند هذه الحالة نكتب مسار الملف كما يلي :
"./data/file.txt"

الحالة الثالثة : ان يكون الملف في المجلد الاب للمجلد الموجود به برنامجنا ، وهنا يصبح مسار الملف كما يلي :
"../file.txt"

الحالة الرابعة : ان نحدد المسار كعنوان انترنت مثل " www.4alarab.com/file.txt "

عموما يجب عليك الا تقلق فهذه المفاهيم العامة سوف تتضح بشكل كبير وسوف تجدها سهلة جدا عندما نتقدم قليلا ونتحدث عن الملفات وكيفية التعامل معها سواء فتح الملفات للقراءة او الكتابة او انشاء الملفات ، لذلك اتوقع منك فقط ان تتعرف فقط على ان مسار الملف مهم كأهمية ان تصف الطريق جيدا لم تريد ان يصل الى موقع محدد ، دون ان يضطر الى سلك الطرق الخاطئة وقد لا يصل ابدا .

اتخاذ القرارات

فـي اثـناء كتابتك للبرنامج سوف تلاحظ انك بحاجة الى اتخاذ قرار معين ، مثلا هـل اسمح للزائر بالدخول الى صفحة معينة او الاطلاع على معلومات محددة ؟ وبالطـبع فان الاجابة على هذا السؤال تعتمد على تحقق شروط معينة مثل هـل يملـك هـذا الشـخص صـلاحيات لدخـول الموقـع وهـل كلمـة السـر التـي ادخلها صحيحة ؟

وهكـذا نحـن بحاجة الى اتخاذ مثل هذه القرارات دائما ولكي نتخذ القرار يجب ان نتأكد من ان الشرط يتحقق مثلا هل كلمة السر صحيحة ، فاذا كانت كلمة السـر صحيحة يعيد لنا البرنامج القيمة true بينما اذا لم تكن صحيحة فسيعيد لنا القيمة false .

اذن الشـرط هو عبارة عن مقارنة بين القيمة التي يدخلها المستخدم والقيمة الموجـودة لديـنا فـاذا كانـت القيمـتان متساويتان فيعني انه ادخل كلمة السر الصحيحة وهذا يعني انه يمكنه الدخول الى الموقع .

الدالة الشرطية IF (اذا كان فـ)

فـي هـذه الدالـة الشـرطية نقـوم بالـتأكد مـن الشــرط وسوف يكون لدينا هنا حالـتان ام ان يـتحقق الشـرط وهـنا سـوف يـتم الدخـول الـى الدالـة وتنفـيذ العمليـات الموجـودة بداخلهـا ثـم الانتقال وتنفيذ الاوامر التي تقع خارجها ، او ان لا يـتحقق الشــرط وهـنا ســوف يـتم الانـتقال مباشــرة الــى نهايـة الدالـة الشرطية وتنفيذ الاوامر التي تقع خارجها .

صيغتها :

```
if ( } ( الشرط
نفذ هذا اذا تحقق الشرط
}
أي اوامر خارج الدالة الشرطية
```

مثال :

```
؟>php
$a=”pass” ;
if ($a=="pass" ) {
echo $a ;
}
print “End” ;
<؟
```

فـي هـذا المـثال نختـبر القـيمة الموجـوده في المتغير $a فاذا كانت تساوي pass نقـوم بطباعـتها ثـم ننـتقل الى خارج دالة القرار ، اما اذا كانت لا تساوي هذه القيمة فاننا ننتقل مباشـرة الى خارج دالة القرار .

تلاحـــظ هـــنا انـــنا نـــنفذ اوامـــر معيـــنه فقـــط عـــندما يـــتحقق الشـــرط .

الدالة الشرطية IF else (اذا كان فـ واذا لم يكن فـ)

فـي هـذه الدالـة الشـرطية نقـوم بالـتأكد مـن الشـرط وسوف يكون لدينا هنا حالـتان ام ان يـتحقق الشـرط وهـنا سـوف يـتم الدخـول الـى الدالـة وتنفـيذ العملـيات الموجـودة بداخلهـا ثـم الانتقال وتنفيذ الاوامر التي تقع خارجها ، او ان لا يـتحقق الشـرط وهنا سوف يتم تنفيذ الاوامر الموجودة بعد الكلمة else ثم الانتقال مباشـرة الى نهاية الدالة الشرطية وتنفيذ الاوامر التي تقع خارجها .

صيغتها :

```
if ( } ( الشرط
نفذ هذا اذا تحقق الشرط
} else {
نفذ هذا اذا لم يتحقق الشرط
}
```

مثال :

```
؟>php
if ($a=="sager" ) {
echo $a ;
} else {
echo " ؛ "كلمة السر غير صحيحة
}
print “end” ;
<؟
```

فـي هـذا المـثال نختـبر القـيمة الموجـوده في المتغير $a فاذا كانت تساوي sager نقـوم بطباعـتها ثم ننتقل الى خارج دالة القرار ، اما اذا كانت لا تساوي هـذه القـيمة فانـنا نـتقل الـى الخـيار الـبديل الذي يأتي بعد الكلمة else ثم ننتقل الى خارج دالة القرار .

تلاحـظ هنا اننا ننفذ اوامر معينه عندما يتحقق الشرط ، وننفذ ايضا اوامر معينه اذا لم يتحقق الشرط .

الدالة الشرطية elseif (اذا كان فـ او اذا كان فـ اواذا لم يكن فـ)

فـي هـذه الدالـة الشـرطية نقـوم بالـتأكد مـن الشـرط وسـوف يكون لدينا هنا حالـتان ام ان يـتحقق الشـرط وهـنا سـوف يـتم الدخـول الـى الدالـة وتنفـيذ العملـيات الموجـودة بداخلهـا ثـم الانتقال وتنفيذ الاوامر التي تقع خارجها ، او ان لا يـتحقق الشـرط وهـنا سـوف يـتم التأكد من الشرط الثاني وهكذا الثالث واذا لم تتحقق الشروط جميعا يتم تنفيذ الاوامر الموجودة بعد الكلمة else ثم الانتقال مباشرة الى نهاية الدالة الشرطية وتنفيذ الاوامر التي تقع خارجها .

صيغتها :

```
if () { الشرط الاول )
نفذ هذا اذا تحقق الشرط الاول
} elseif () { الشرط الثاني )
نفذ هذا اذا تحقق الشرط الثاني
} else {
نفذ هذا اذا لم يتحقق أي من الشروط
}
```

مثال :

```
?>php
if ($a=="sager" ) {
echo "؛ "اهلا صقر
} elseif ($a=="khaled") {
echo "؛ "اهلا خالد
} else {
echo " ؛ "الاسم غير معروف
}
<?
```

فـي هـذا المـثال نختـبر القـيمة الموجـوده في المتغير $a فاذا كانت تساوي sager نقوم بطباعة رسالة الترحيب اهلا صقر ثم ننتقل الى خارج دالة القرار ، بـينما اذا كانـت القـيمة تسـاوي khaled فاننا نطبع رسالة الترحيب اهلا خالد ثـم نـنـتقل الـى خـارج دالـة القـرار ، امـا اذا كانـت القـيمة لاتسـاوي جمـيع الاحـتمالات فانـنا ننتقل الى الخيار البديل الذي يأتي بعد الكلمة else ونطبع الجملة الاسم غير معروف ثم ننتقل الى خارج دالة القرار .

تلاحـظ هـنا انـنا نـنفذ اوامـر معيـنه عندما يتحقق احد الشروط او الاحتمالات ، ونـنفذ ايضـا اوامـر معيـنه اذا لـم يـتحقق أي مـن هذه الشروط او الاحتمالات .

حلقة التكرار while

استمر في حلقة مادام الشرط متحققا ويتم التحقق من الشرط قبل الدخول الى الحلقة في كل مرة ، فعند بداية الحلقة نتأكد من الشرط فاذا تحقق نقوم بالدخول الى الحلقة وتنفيذ الاوامر مرة واحدة ثم نعود لرأس الحلقة مرة اخرى ونتأكد من الشرط وهكذا يستمر البرنامج حتى يصبح الشرط غير متحقق فيقوم بالخروج من الحلقة وتنفيذ الاوامر الموجودة خارجها .

وتساعد الحلقات كثيرا في اختصار الاوامر التي تكتب فلو اردنا ان نحصل على مائة سجل من قاعدة البيانات وطباعتها فاذا لم نستخدم الحلقة فسنحتاج لمئات الاوامر ولكن بواسطة الحلقة المناسبة فلن نكتب اكثر من عشرة اسطر .

صيغتها :

```
while ( الشرط ):
نفذ دائما مادام الشرط متحققا
endwhile;
```

مثال :

```
<?php
$a=0 ;
while ($a <18 ):
++$a ;
echo $a ;
endwhile;
?>
```

في هذا المثال سوف نرى مفعول الحلقة في اختصار كتابة الاوامر ففي هذا المثال البسيط نحن بحاجة الى طباعة الارقام من ١ الى ١٧ فلو استخدمنا الطريقة العادية فسوف نحتاج الى ١٧ سطر يحتوي على جملة echo يليها المتغير الذي نود طباعته وبالاضافة الى زيادة الكوود يكون من الصعب صيانة البرنامج او تطويره .

فلو اردنا مثلا طباعة الارقام حتى ١٢٠ فكل ماعلينا هو تغيير الرقم الموجود في السطر الثالث من ١٨ الى ١٢٠ وهكذا .

لا حظ ان البرنامج يتحقق من صحة الشرط قبل دخوله للحلقة ومن ثم فاننا لن نطبع القيمة ١٨ لانها تخالف الشرط .

حلقة التكرار Do while

استمر في حلقة مادام الشرط متحققا ويتم التحقق من الشرط في نهاية الحلقة في كل مرة ،ففي البداية ننفذ الاوامر اول مرة دون الرجوع للشرط وعند نهاية الحلقة نتأكد من الشرط فاذا تحقق نعود الى الحلقة وننفذ الاوامر مرة اخرى ونتأكد من الشرط وهكذا يستمر البرنامج حتى يصبح الشرط غير متحقق فيقوم بالخروج من الحلقة وتنفيذ الاوامر الموجودة خارجها .
وهنا نلاحظ ان هذه الحلقة شبيهه بالحلقة السابقة ولكن الفرق ان الحلقة السابقة تتأكد من الشرط عند بداية الحلقة أي انها لاتسمح بالدخول الى الحلقة الا بعد تحقق الشرط اما الثانية فتنفذ الاوامر مرة واحده وبعدها تتأكد من الشرط .

صيغتها :

```
do {
نفذ دائما مادام الشرط متحققا
} while (الشرط)؛
```

مثال :

```
<؟php
$i = 0 ;
do {
  ++$i;
  print $i;
} while ($i<18);
؟>
```

في هذا المثال البسيط نحن بحاجة الى طباعة الارقام من ١ الى ١٨ فلو استخدمنا الطريقة العادية فسوف نحتاج الى ١٨ سطر يحتوي على جملة echo يليها المتغير الذي نود طباعته وبالاضافة الى زيادة الكوود يكون من الصعب صيانة البرنامج او تطويره .
فلو اردنا مثلا طباعة الارقام حتى ١٢٠ فكل ماعلينا هو تغيير الرقم الموجود في السطر السادس من ١٨ الى ١٢٠ وهكذا .
الاختلاف بين هذه الطريقة والطريقة السابقة انها تتحقق من الشرط في نهاية الحلقة ولذلك سوف تلاحظ انها تطبع القيمة ١٨ رغم انها تخالف الشرط .

قد تسأل ما الحاجة لها اذن ؟
اولا يجب ان تعرف انك تستطيع الوصول الى هدفك بطرق مختلفه ولكن دائما الطريقة المختصرة هي الافضل والاسهل صيانة ولذلك تم ايجاد الدوال المختلفة لتستفيد منها في برامجك المختلفة .

حلقة التكرار for

استمر في حلقة مادام الشرط متحققا ويتم التحقق من الشرط في بداية الحلقة ولكن هنا نحن نحدد طريقة تصرف الشرط أي نحن نحدد طول الحلقة عن طريق تحديد بدايتها وشرط الحلقة وطريقة تصرف الحلقة .

صيغتها :

```
for ( بداية الحلقة; شرط الحلقة ؛ طريقة تصرف الحلقة) {
نفذ دائما مادام الشرط متحققا
}
```

مثال :

```
<?php
for ($i = 1; $i < 10; $i++) {
 print $i;
 }
?>
```

في هذا المثال وضعنا بداية الحلقة بمتغير قيمته ١ ووضعنا شرط الحلقة بحيث تستمر الحلقة مادامت قيمة المتغير اقل من عشرة ويتزايد بشكل متصاعد وهكذا سوف يتم طباعة القيم من ١ الى ٩ .

حلقة التكرار each او foreach

تعتمد هـذه الحلقة على وجود كائن تبحث في عناصره مثل المصفوفة بحيث نسـتمر فـي الحلقة مـادام هـناك عناصـر موجـودة مـن الكائـن المحـدد مـثل الجـداول او الحقـول وهـذه الحلقـة اروع الحلقـات لتسـهيلها العمـل مـع المصـفوفات التـي تنشـئها او التي تحصل عليها عن طريق دوال البرنامج مثل الحصـول علـى مصـفوفة بها اسـماء جميع الموظفين في القسم المحاسبي .

صيغتها :

هناك الكثير من الصيغ الممكنة لهذه الدالة ومن اشهرها :

```
foreach ( متغير as كائن) {
نفذ دائما مادام هناك عناصر
}
```

مثال :

```
<?php
$a = array (2, 4, 6, 8);
foreach ($a as $v) {
  print $v ;
}
?>
```

فـي السـطر الثانـي من هذا المثال أنشئنا مصفوفة ووضعناها في المتغير $a فـي الامثلة العملية قد تكون المصفوفة قادمة من قواعد البيانات او دالة اخرى ، ثـم فـي السـطر الثالـث طلبـنا مـن الـبرنامج سـحب العناصـر مـن المصفوفة ووضـعها فـي المتغيـر $v عنصـراً بعـد آخـر ، وفـي السـطر الـرابع طلبـنا مـن الـبرنامج طباعة المتغير الجديد والنتيجة سوف تكون طباعة جميع الاعداد في المصفوفة .

الدالة break

عـندما تكـون فـي احد الحلقات وتريد الخروج من هذه الحلقة قبل ان تنتهي ، مـثلا لـو كنـت تـتعامل مـع قواعـد البـيانات و اردت فقـط اول عشــرة ســجلات فســتحتاج لهـذه الدالـة التـي تجعلـك تقفـز مباشــرة الـى خـارج الحلقـة عـند حصولك على البيانات .

صيغتها :
للخروج من الحلقة فقط اكتب الامر التالي :

```
break;
```

مثال :

```
؟>php
$a = array (2, 4, 6, 8);
foreach ($a as $v) {

if ($v == 6) {
break;
}

  print $v ;

}
<؟
```

هـذا المـثال شــبيه للمـثال الســابق ففي السطر الاول من هذا المثال أنشئنا مصـفوفة ووضـعناها فـي المتغـير $a ثم في السطر الثاني طلبنا من البرنامج سـحب العناصـر من المصفوفة ووضعها في المتغير $v عنصراً بعد آخر ، وفي السـطر الثالـث طلبـنا مـن الـبرنامج الـتأكد من العنصر هل يساوي ٦ فاذا كان كذلك عليه الخروج من الحلقة واذا لم يكن يقوم بطباعته والنتيجة سوف تكون طباعة الرقمين ٢ و ٤ فقط .

الدالة الشرطية switch

عـندما يكون لدينا متغير مثلا اسم المستخدم ونريد ان نتأكد هل هو محمد ام خـالد ام سـعد ام فهـد فعليـنا اسـتخدام الدالـة if و elseif وتكـرارها بشـكل ممـل ويجعـل مـتابعة الـبرنامج فـيما بعـد متعـبة جـداً ولذلـك مـن الأفضـل هنا استخدام الدالة switch التي سوف تسهل العمل بشكل كبير كما سنرى .

صيغتها :

```
switch ( } ( المتغير
   case :الاحتمال الاول
      المطلوب لهذا الاحتمال
   case :الاحتمال الثاني
المطلوب لهذا الاحتمال
   case ....... :الاحتمال
المطلوب لهذا الاحتمال
}
```

مثال :

```
?>php
    $a="خالد" ؛
switch ($a) {

   case "فهد":
  echo " اهلا فهد كلمة السر لك هي ١١١١" ؛
   break;

   case "محمد":
  echo " اهلا محمد كلمة السر لك هي ٢٢٢٢ " ؛
  break;

   case "خالد":
  echo " ٣٣٣٣ اهلا خالد كلمة السر لك هي" ;
   break;

   case "سعد":
  echo " اهلا سعد كلمة السر لك هي ٤٤٤٤ " ؛
   break;
```

```
}
?>
```

في هذا المثال وضعنا في المتغير $a افتراضيا الاسـم خالد ومن ثم طلبنا من الـبرنامج ان يسـتخدم الدالـة switch ليـتعرف على اسـم المسـتخدم ويخبره عن كلمة السر الخاصة به .

ملاحظة : من المهم جداً ان نضع الدالة ;break بعد نهاية كل احتمال .

الدالة require و include

عندما يكون لدينا برنامج كبير فلا اعتقد انه من الحكمة وضع جميع الاوامر في ملف واحد ، بل سنقوم بتقسيم البرنامج الى عدة ملفات ، مثلا ملف للتأكد من اسم المستخدم وملف للتسجيل وملف للاتصال مع قاعدة البيانات وهكذا ، ولكي نستدعي هذه الاجزاء عند الحاجة نحتاج لاحد هاتين الدالتين .
وعند استدعاء أي ملف فسوف تتنفذ الاوامر الموجودة داخلة وسوف نستطيع استخدام المتغيرات والدوال الموجودة فيه كانها في ملفنا الاصلي .

صيغتها :

```
require ( اسم الملف )؛
include ( اسم الملف )؛
```

مثال ١ :

```
require (“file.php”);
```

مثال ٢ :

```
include (“file.php);
```

في هذين المثالين من خلال هذا الامر نستدعي الملف file.php وننفذ جميع الاوامر الموجودة فيه ، بل ان المتغيرات الموجودة فيه نستطيع التعامل معها كما لو كانت في الملف الاصلي .
ملاحظة : نستطيع ان نكتب اسم الملف فقط اذا كان في نفس المجلد ولكن اذا كان في مجلد آخر فيجب ان نكتب المسار الصحيح للملف .

الدوال Function

في هذا الجزء سوف نتحدث عن كيفية انشاء دوال خاصة بك داخل البرنامج ولكن في البداية دعنا نتعلم ماهي الدوال ولماذا نحتاجها .
الدالة هي مجموعة من الاوامر والمتغيرات تكون بشكل مستقل داخل البرنامج ولا كن البرنامج يتجاهلها ولا ينفذها الا اذا تم استدعائها عن طريق كتابة اسم الدالة وارسال المتغيرات المطلوبة لها .
وقد تتسائل هنا ما اهمية هذه الدوال او لماذا لا اكتب الاوامر بدون الدخول الى هذه المتاهة ؟ والجواب انك سوف تجد ان هذه الدوال سوف توفر لك الكثير من الوقت بالاضافة الى جعل برنامج مرتب وسهل التطوير ، فمثلا عندما نحتاج ان نتأكد من اسم المستخدم وكلمة المرور في برنامجنا وهذه سوف نحتاج اليها عشرات المرات في برامج قواعد البيانات فكل ماعلينا هو كتابة دالة نرسل لها الاسم وكلمة السر لتتأكد من ان هذا المستخدم لديه الصلاحيات في الدخول واجرء التغيرات .

ماهي الصيغة التي نكتب بها الدوال

كل دالة يجب ان تبدأ بالكلمة function ثم اسم الدالة وهو اسم تختاره انت ثم تكتب المتغيرات التي تود ان ترسلها للدالة وبعد هذا تكتب العمليات التي تريد من الدالة القيام بها ، وفي النهاية سوف تعيد لك الدالة متغير نتيجة هذه العمليات ويجب ان تسبق المتغير المعاد الكلمة return .
حسب ماشرحنا فان صيغة الدالة العامة هي :

Function } (المتغيرات التي سوف ترسل للدالة) اسم الدالة
هنا الاوامر والعمليات
return القيمة المعادة ;
} وهذه نهاية الدالة {

مثال :

```
؟>php
Function AddNo ($no1 ,$no2) {
$no3 = $no1 + $no2 ;
return  $no3 ;
}

print AddNo (5,3) ;
print AddNo (4,2) ;
```

```
?>
```

في هذا المثال انشئنا دالة اسمها AddNo وتستقبل متغيرين وتقوم بعملية جمع المتغير الاول مع الثاني وتعيد لنا النتيجة ، وتلاحظ اننا استدعينا الدالة مرتين الاولى ارسلنا لها رقمين هما ٥ و ٣ وسوف يعيد لنا الرقم ٨ وفي المرة الثانية ارسلنا لها الرقمين ٤ و ٢ وسوف يعيد لنا الرقم ٦ ، وايضا تلاحظ اننا كتبنا الدالة قبل امر الاستدعاء لكي تتأكد فقط ان البرنامج لا ينفذها الا اذا تم استدعائها فلو مسحت سطر الاستدعاء فلن يحدث شيء عند تنفيذ البرنامج .

اسئلة واجوبة حول الدوال :

- في المثال السابق يحدث خطأ اذا قام المستخدم باستدعاء الدالة دون ان يرسل لها قيم او ارسل لها قيمة واحدة فقط فهل يمكن ان اجعل الدالة تضع قيمة افتراضية اذا لم قام المستخدم بأرسال قيم فارغة ؟

نعم يمكن وضع قيم افتراضية للمتغيرات التي تستقبلها الدالة وهذه القيم تستخدمها الدالة فقط في حالة لم يقم المستخدم بأرسال أي قبم واليك المثال :

```
?>php
Function AddNo ($no1=0 ,$no2=0) {
$no3 = $no1 + $no2 ;
return  $no3 ;
}

print AddNo () ;
print AddNo (4) ;
print AddNo (4,2) ;
<?
```

في هذا المثال وضعنا قيم افتراضية للمتغير الاول والثاني وهي القيمة صفر ثم استدعينا الدالة ثلاث مرات الاولى بدون ارسال أي قيم فكانة النتيجة صفر ، والثاني بإرسال القيمة ٤ فكانة النتيجة اربعة ، واخيرا ارسلنا قيمتين فكانت النتيجة حاصل جمعهما .

- في المثال السابق اريد ان ارسل قيمتين و أود ان تعيد لي حاصل الجمع والضرب فهل اعمل دالة جديدة ام يمكن ان تعيد لي الدالة اكثر من قيمة ؟

في الحقيقة الدالة تعيد قيمة واحدة فقط ولكن نستطيع التغلب على هذا بواسطة عدة طرق من اهمها ان نجعل الدالة تعيد لنا مصفوفة من القيم ، فلكي نحقق الهدف نحتاج ان نجعل الدالة تعيد لنا مصفوفة تتكن من رقمين الاول حاصل الجمع والثاني حاصل الضرب والعملية في غاية السهولة واليك المثال :

```
؟>php
Function AddNo ($no1=0 ,$no2=0) {
$no3 = $no1 + $no2 ;
$no4 = $no1 * $no2 ;

return  array ($no3 ,$no4) ;
}

  list($back1,$back2 )=AddNo (2,6) ;
print $back1 ;
print $back2 ;
<?
```

فـي هـذا المـثال جعلـنا الـبرنامج يعـيد لنا مصفوفة تحتوي على حاصل الجمع والضـرب وعـند اسـتدعائنا للمصـفوفة طلبـنا وضـع القـيم القادمة في متغيرين الاول $back1 والثانـي $back2 وبعـد ذلـك يمكـن الـتعامل مع هذه المتغيرات كما نشاء وهنا اخترنا طباعتها .

- فـي المـثال السـابق اود ان احصـل علـى المتغـيرات بـدون اسـتخدام return ولا المصـفوفات فقـد تعلمـت فـي المتغـيرات ان بأمكانـي ان اجعل المتغير عام فكيف افعل ذلك هنا ؟

مـع انني افضل دائما استخدام المصفوفات لما فيها من المرونة والقوة ولكني الاحـظ ان الكثـير مـن المبرمجيـن يسـتخدمون المتغـيرات العامـة بكـثره داخـل الـدوال ولكـل رأيـته الخاصـة فـي هـذا المجال ولذلك رأيت ان اجعل الطريقتين معروفتيــن لديــك وتســتخدم ماتشــاء والــيك نفــس المــثال الســابق ولكــن بالمتغيرات العامة :

```
؟>php
Function AddNo ($no1=0 ,$no2=0) {
$GLOBALS["no3"] = $no1 + $no2 ;
$GLOBALS["no4"] = $no1 * $no2 ;

}
```

```
addno (2,3) ;
print $no3  ;
print $no4 ;
<?
```

ارأيت لقد حللنا المسألة بطريقة مختلفة وهذه حال البرمجة فقد تكتب برنامج مـن عشـرين سـطر ويأتى اخر ويكتب نفس البرنامج من مئة وخمسـين سـطر ! هـل تعـتقد انـي ابالغ ؟ كما تشـاء ولكني رأيت ذلك بنفسـي ولم يحدثني عنه احد .

التعامل مع التاريخ

تستطيع بواسطة php الحصول على تاريخ اليوم بل اكثر دقة تستطيع ان تطلبه منه ان يعطي لك اسم اليوم والساعة والدقيقة والثانية وهل الوقت صباحا او مساءا وهل تريد ان تحصل على اسم الشهر مختصرا (sep) او كاملا (September) .

ولكي تحصل على التاريخ والوقت هناك دالتان لهذا الغرض وهما :
الدالة date والدالة gmdate وكلاهما يقومان بنفس المهمة و يأخذان نفس المتغيرات ولكن الاختلاف الوحيد بينهما ان الدالة gmdate تحاول الرجوع بالوقت على حسب التوقيت العالمي (توقيت غرينتش) فتقرأ اعدادات السيرفر فاذا كنت مثلا في المملكة العربية السعودية فسوف تقوم بطرح ٣ ساعات من الوقت للوصول الى الوقت في مدينة غرينتش .
عموما سوف استخدم في الامثلة الدالة date التي تعطي الوقت في السيرفر وجميع الامثلة تنطبق على الدالة gmdate بدون استثناء .

لكي تحصل على الوقت من الدالة date عليك كتابة الصيغة التالية :

Date("ماذا تريد ")

فلو اردنا ان نعرف السنة الان فاننا سوف نكتب :

Date("Y")

لاحظ اننا كتبنا حرف Y بشكل كبير اذن سوف تعيد لنا الدالة السنة كاملة .
عموما الدالة date تأخذ المتغيرات التالية :

المتغير	يعود بـــــــ
Y	يعود بالسنة من اربعة ارقام مثل ١٩٩٩
y	يعود بالسنة من رقمين مثل ٩٩
w	يعود برقم اليوم من الاسبوع الاحد=٠ الاثنين =١ وهكذا حتى نصل الى السبت = ٦
t	يعود بعدد ايام الشهر الحالي
s	يعود بالثواني
S	يعود بامتداد ارقام اليوم th او nd
U	يعود بالثواني بالجزء من الثانية
z	تعود برقم اليوم من السنة
F	تعود بأسم الشهر كاملا مثل September
M	تعود بأسم الشهر مختصراً مثل Sep

m	تعود برقم الشـهر من ٠١ – ١٢
n	تعود برقم الشهر من ١-١٢
j	تعود برقم اليوم من الشهر من ١- ٣١
l	(حرف L صغير) تعود باسـم اليوم كاملا مثل Wednesday
i	تعود بالدقائق من ٠٠ - ٥٩
G	تعود بالساعة بنظام ٢٤ ساعة من ١ – ١٢
g	تعود بالساعة بنظام ١٢ ساعة من ١ - ٢٤
D	تعود بأسـم اليوم مختصرا مثل Wed
d	تعود برقم اليوم من الشـهر من ٠١ – ٣١
a	تعود بـ am او pm
A	تعود بـ AM او PM

مـن دمـج القـيم الموجودة في الجدول السـابق نسـتطيع الحصول عدة اشـكال من التاريخ والوقت حسـب حاجتنا واليك بعض الامثلة :

- للحصول على اليوم والشـهر والسـنة بالارقام نكتب التالي :
-

```
print date  ("j/n/Y") ;
```

وسـوف تكون النتيجة مشابهه لـ (22/3/2002)

- للحصول على الساعة والدقائق والثواني بالاضافة الى AM او PM نكتـــــب التالي :

```
print date  ("h:i:s A");
```

وسـوف تكون النتيجة مشابهه لـ (05:11:19 AM)

وهكذا تسـتطيع الحصول على التشـكيلة التي تريد مهما كان طبعها .

مثال متقدم

مـاذا لـو اردنـا طـباعة اسـم الـيوم بالعربـي أي السـبت او الاحـد وهكـذا كـيف نسـتطيع ذلك ؟
لحـل هـذه المسـألة نعـود للجـدول السـابق لنجد ان المتغير w يعيد لنا رقم الـيوم فـي الاسـبوع وهـنا نرى هذا المتغير كم قيمته فاذا كانت = ٠ فهو يوم الاحـد امـا اذا كانـت = ١ فهـو يـوم الاثنيـن وهكذا بقية الايام وسـوف نسـتخدم الدالة الشرطية switch التي شرحناها من قبل عموما اليك الحل :

```
<؟

    $a= date("w");
switch ($a) {

    case 0:
  echo "الاحد"؛
   break;

    case 1:
  echo "الاثنين"؛
   break;

    case 2:
  echo "الثلاثاء"؛
   break;

    case 3:
  echo "الاربعاء"؛
   break;

    case 4:
  echo "الخميس"؛
   break;

    case 5:
  echo "الجمعة"؛
   break;

    case 6:
  echo "السبت"؛
   break;

}

؟>
```

التعامل مع الملفات

نستطيع بواسطة php القيام بالكثير من العمليات على الملفات والمجلدات ومنها على سبيل المثال نسخ ملف او حذف ملف او انشاء ملف او انشاء مجلد او حذف مجلد او معرفة نوع الملف او غيرها من العمليات التي يتيحها لنا php وسوف نتعرف هنا على الدوال التي تساعدنا على القيام بهذه العمليات مع ذكر مثال على كل دالة .

الدالة basename

تقوم هذه الدالة باستخلاص اسم الملف من مسار معين ، فلو كان لدينا المسار :

4alarab.com/help/index.php

واردنا ان نعرف اسم الملف فاننا نمرره الى هذه الدالة كما يلي :

```
echo basename("4alarab.com/help/index.php") ;
```

وسوف تكون النتيجة طباعة اسم الملف index.php .

الدالة dirname

في الدالة السابقة تعرفنا على كيفية استخلاص الاسم والان سوف نتعرف على كيفية استخلاص اسم المجلدات من المسار ، فلو كان لدينا المسار :

4alarab.com/help/index.php

واردنا معرفة المجلدات التي تقود الى هذا الملف فاننا سوف نكتب التالي :

```
echo dirname  ("4alarab.com/help/index.php");
```

وسوف تكون النتيجة 4alarab.com/help .

الدالة chmod

عندما نرسل الملفات الى الموقع او المجلدات فاننا نحتاج في بعض الاحيان تعديل التراخيص لهذه الملفات او المجلدات سواء ان نتيحها للكتابة والقراءة او

نجعلها فقط للقراءة ، وعموما لو استخدمت برامج ftp في ارسال ملفاتك الى الانترنت فانك سوف تعرف ان اهم ترخيصين هما ٧٥٥ و ٧٧٧ .
الاول ٧٥٥ يعطى لملفات السكربتات ، والثاني ٧٧٧ يعطى للملفات و للمجلدات التي يراد اتاحتها للكتابة .
ومن هنا لكي نقير الترخيص لملف او مجلد نكتب الامر التالي :

```
chmod ("test", 0755);
```

هنا سوف نعطي الترخيص ٧٥٥ للمجلد test ولاحظ اننا اضفنا الصفر في بداية الترخيص ، ولو اردنا اعطاءه الترخيص ٧٧٧ فسوف نكتب ٠٧٧٧ وهكذا لبقية التراخيص.

الدالة chown

في بعض المواقع يكون هناك اكثر من مستخدم للسيرفر مما يعني ان بعض الملفات او المجلدات تكون ملك هذا المستخدم او ذاك وقد تحتاج الى اعطاء هذه الملكية او تغييرها في وقت من الاوقات ولكي تفعل ذلك تحتاج الى هذه الدالة .
ملاحظة : فقط المدير او مايطلق عليه (superuser) هو من يستطيع استخدام هذه الدالة .
chown (الملف او المجلد ، معرف المستخدم) ؛

الدالة copy

كثيرا ما نحتاج الى انشاء نسخة احتياطية لملف ما يحتوى على كلمات سر او معلومات مهمة لكي نستخدمه عند حدوث أي مشكلة ولكي نفعل ذلك نحتاج الى استخدام الدالة copy كما يلي :

```
copy("arab1.jpg", "arab1.pak");
```

في هذا المثال نسخنا الملف arab1.jpg الى الملف arab1.pak في نفس المجلد .

الدالة unlink

من الاعمال الاساسية في عالم الكمبيوتر حذف الملفات او المجلدات فعندما لا نحتاج لملف او مجلد معين فاننا نقوم بحذفه ففي تركه استهلاك من مساحة القرص الصلب ، ولكي تحذف ملف من السيرفر عن طريق برنامجك اليك الامر التالي :

```
unlink ("arab1.pak");
```

ملاحظة : يجب ان يكون الملف arab1.pak في نفس المجلد مع برنامجك او ان تكتب المسار الكامل والصحيح لهذا الملف .

الدالة diskfreespace

هل تود الحصول على المساحة المتبقية في القرص الصلب لديك بالبايت كل ماعليك ان تكتب الامر التالي :

```
echo diskfreespace ("/");
```

وهنا سوف يتم طباعة المساحة المتبقية بالبايت ، ماذا لو كان لديك مجلد له حصة معينة او دومين فرعي تود ان تعرف المساحة المتبقية له ، كل ماعليك هو كتابة اسم المجلد لتحصل على المساحة بالبايت .

الدالة file

نحتاج لقراءة الملفات واستخلاص البيانات منها وتعتير هذه الدالة من الدوال المتميزه في هذا المجال فهي تحتاج فقط الى ارسال اسم الملف اليها ثم تقوم بقراءة الملف كاملا ووضعه سطرا سطرا داخل مصفوفة ، بحيث يصبح كل سطر في متغير من متغيرات المصفوفة .
وسوف نتعرف على مثال كامل للقراءة والكتابة من والى الملفات في الصفحات القادمة ولكن هنا دعنا نتعرف على صيغة هذا الامر :

```
$myarry = file ('http://www.4alarab.com/index.html');
```

وكما تلاحظ فهذه الدالة لا تحتاج الا الى اسم الملف لتقرأه لك وتضعه في المصفوفة التي تحددها والتي سميناها هنا myarry .

الدالة file_exists

لكي نتعامل مع ملف سواء بالقراءة او الكتابة او غيرها لابد من معرفة هل الملف موجود في الموقع السليم ام لا ، لاننا اذا حاولنا ان نتعامل معه ولم يكن موجود سوف نقع في اخطاء يجب ان لا يقع فيها مبرمج لديه القدر القليل من المعرفة البرمجية ، ولذلك لدينا هذه الدالة file_exists التي نرسل

لها اسم الملف فتعيد لنا القيمة ١ اذا كان الملف موجود او أي قيمة اخرى اذا لم يكن موجود .
ولنأخذ هذا المثال :

```
if (file_exists("pic.gif")){
echo "الملف موجود "؛
} else {
echo "الملف غير موجود "؛
}
```

في هذا المثال في السطر الاول نختبر هل الملف موجود فاذا كان موجود نعرض الرسالة الملف موجود واذا لم يكن الملف موجود تعرض الرسالة الملف غير موجود .

الدالة filesize

اثناء التعامل مع الملفات قد تحتاج الى معرفة حجم الملف بالبايت وهذه الدالة تعود لك بذلك ، كل ماعليك ارسال اسم الملف للدالة لتحصل على الحجم :

```
echo filesize ("pic.gif") ;
```

في هذا المثال نطبع حجم بالبايت .

الدالة filetype

هذه الدالة تعيد نوع الملف فعند ارسال اسم الملف تعود بنوع الملف ، والدالة تعيد احد هذه القيم (fifo, char, dir, block, link, file, and unknown) واليك المثال :

```
echo filetype ("pic.gif") ;
```

في هذا المثال نقوم بطباعة نوع الملف .

الدالة Fopen

لكي نتعامل مع أي ملف لابد ان نقوم بفتحه وتوفر لغة php الكثير من الدوال لهذا العملية ، ولا نتعجب من ذلك فكل برنامج نقوم به نجده يتعامل مع الملفات بطريقة ما .

عموما سنتطرق لهذه الدوال جميعا بالتفصيل في القسم امثله على التعامل مع الملفات بعد هذا الفصل .

تحتاج هذه الدالة الى ثلاث متغيرات اثنان يجب ارسالهما والثالث اختياري أي يمكن الاكتفاء بمتغيرين فقط وهذا مايستخدم غالبا وتعود اليك برقم يدعى مقبض الملف عن طريقه تتعامل مع الدوال الاخرى واليك المتغيرات :

المتغير الاول : اسم الملف المراد فتحه .

المتغير الثاني : ماهي الحالة التي تريد ان تفتح الملف عليها (للقراءة فقط / للقراءة والكتابة / او للكتابة فقط) .

المتغير الثالث : وهو اختياري ويحدد هل تريد استخدام المجلدات المحددة مسبقا في خيارات php ، القيمة ١ تعني نعم .

المتغيرات التي تعبر عن حالات فتح الملف :

المتغير	العمل
r	**فتح الملف للقراءة فقط ووضع المؤشر في البداية**
r+	**فتح الملف للقراءة والكتابة ووضع المؤشر في البداية**
w	**فتح للكتابة فقط ووضع المؤشر في البداية ومسح جميع البيانات ، اذا لم يكن الملف موجود ستقوم الدالة بأنشاءه**
w+	**فتح الملف للقراءة والكتابة فقط ووضع المؤشر في البداية ومسح جميع البيانات ، اذا لم يكن الملف موجود ستقوم الدالة بأنشاءه**
a	**فتح للكتابة فقط ووضع المؤشر في النهاية ، اذا لم يكن الملف موجود ستقوم الدالة بأنشاءه**
a+	**فتح الملف للقراءة والكتابة فقط ووضع المؤشر في النهاية ، اذا لم يكن الملف موجود ستقوم الدالة بأنشاءه**

واليك هذه الامثلة :

```
$fp = fopen ("file.txt", "r");
$fp = fopen ("file.txt", "w");
$fp = fopen ("http://www.4alarab.com/file.txt", "r");
$fp = fopen ("http://www.4alarab.com/file.txt", "w");
```

الدالة fclose

كما تحدثنا في الدالة السابقة فاننا عندما نقوم بفتح أي ملف فأننا سوف نحصل على رقم نسميه مقبض لهذا الملف نتحكم بواسطته بهذا الملف مادام مفتوحا ، وايضا مادام هذا الملف مفتوحا فاننا نستهلك جزءا من ذاكرة الجهاز (السيرفر) ولذلك علينا عند الانتهاء من هذا الملف ان نقوم بإغلاقه وعندما نفعل ذلك فسوف تتحرر الذاكرة التي كانت تستخدم للتعامل معه .
ومع اهمية الدالة فأن استخدامها سهل جدا فكل ماعليك ان تفعل لاغلاق ملف معين ان ترسل مقبضه الى هذه الدالة كما يلي :

```
$fp = fopen ("file.txt", "r");
Fclose ($fp) ;
```

وكما تعلم فالمقبض $fp حصلنا عليه عند فتح الدالة .

الدالة fread

لقد تعرفنا في الدالتين السابقتين على فتح الملف واغلاقه ، ولكن عند فتح الملف فكيف نقرأ البيانات الموجوده فيه ؟
هناك دوال عديدة تقرأ البيانات وهذه من اهمها ، وتتميز هذه الدالة بأنها تقرأ جزء معين من الدالة يحدده المبرمج فقط مما يوفر في الذاكره ، فلو كان الملف المراد قراءته كبير وقمنا بقراءته مرة واحدة فسوف يستهلك جزء كبير من الذاكرة بل قد يؤثر على عمل النظام ككل ، ولذلك فمن الأفضل في هذه الحالة قراءة الملف كأجزاء صغيرة .
وهنا تبرز ميزة الدالة فلو كان لدينا ملف طوله مائة بايت مثلا وفتحناه ثم قمنا بقراءته بواسطة هذه الدالة وحددنا لها الحجم بـ ٢٠ بايت فسوف تحضر لنا اول ٢٠ بايت ، وعند استدعائها مرة اخرى على نفس الملف فسوف تحضر الـ ٢٠ بايت التالية وهكذا حتى نصل نهاية الملف .
اذن من حديثنا السابق فالدالة تحتاج الى متغيرين الاول مقبض الملف والثاني الحجم الذي نريد منها قراءته .

والان اليك هذا المثال :

```
$fp = fopen ("file.txt", "r");
$contents = fread ($fd, 20);
fclose ($fd);
```

ملاحظة : قد تود قراءة الملف كاملا بواسطة هذه الدالة وبهذه الحالة فأنت بحاجة الى معرفة حجم الملف بالتحديد لترسله الى الدالة ، واعتقد انك تعلم ان الامر بسيط جدا فقد تعرفنا سابقا على الدالة التي تعيد لنا حجم الملف

وهـي الدالة filesize وفيما يلي المثال السـابق ولكن بقراءة كافة الملف مرة واحدة :

```
$fp = fopen ("file.txt", "r");
$contents = fread ($fd, filesize("file.txt" ));
fclose ($fd);
```

الدالة feof

هـذه الدالـة خفـيفة وبسـيطه ولكـنها مهمـة فـي الـتعامل مـع الملفات وهي تفيدنا هل تم قراءة الملف بالكامل أي هل وصلنا الى نهاية الملف .
ولاسـتخدام هـذه الدالة كل ماعلينا هو ارسال مقبض الملف وسوف يعيد الينا القيمة true اذا كنا وصلنا الى نهاية الملف .

```
feof ($fp)
```

الدالة fwrite

اذا تحدثـنا عـن القـراءة مـن الملفـات فبالطبع سوف نتحدث عن الكتابة اليها ، وهـنا لديـنا هـذه الدالـة التـي سـوف تقـوم بهذه العملية بشرط ان ترسل لها المتغيرات التالية :

المتغير الاول : (يجب ارساله) وهو مقبض الملف المراد الكتابة عليه .
المتغير الثاني : (يجب ارساله) وهو النص المراد كتابته .
المتغـير الثالـث : (اختياري) وهو حجم النص المراد كتابته وغالبا لا يستخدم .

وقبل ان نأخذ مثال على طريقة استخدام هذه الدالة لك الحق ان تتساءل اين سـيتم كتابة النص المرسـل للمف ؟ هل هو في البداية ام النهاية ام المنتصف ؟
والجـواب هـو : انـه سـوف يتم الكتابة في الموقع الذي توقف فيه مؤشر قراءة الملـف وهـو ماسـوف نـتعرف علـيه عـندما نـأخذ امـثله مفصـله علـى القـراءة والكتابة من الملفات .

واليك المثال الذي يبين طريقة استخدام هذه الدالة :

```
Fwrite($fp , "MyProgram") ;
```

الدالة file

هـذه الدالـة العملاقـة وهي محبوبة المبرمجين فهي تقرأ الملف بشكل كامل وتضعـه فـي مصـفوفة بحيث يمثل كل سطر عنصر في هذه المصفوفة ، وايضا هـذه الدالـة لا تحـتاج الـى مقـبض الملـف بل تحتاج فقط الى اسم الملف أي لانحتاج الى استخدام دالة اخرى لفتح الملف قبل قراءته ، ولا دالة الى اغلاق الملف بعد الانتهاء منه .

مع كل هذه المميزات الفريدة من نوعها في هذه الدالة فقد تتساءل لماذا ازعجتك بالحديث عن كل هذه الدوال السابقة ؟
الجواب ذكرته سابقا فأولا الموارد من الذاكرة لدينا دائما قليلة ولو اردنا قراءة ملف كبير وكنت عملة البرنامج لشركة فسوف يعتذرون لك بأدب عن استخدام برنامجك رغم مميزاته لانه يستهلك الذاكرة ويعطل العمل لديهم .
هل فهمت لماذا تحدثنا عن الدوال الاخرى ؟ عموما هذه الدالة تعتبر متميزة جدا في الملفات ذات الاحجام المتوسطة .
والان اليك مثال لكيفية استخدامها :

```
<?php

$fcontents = file ('http://www.4alarab.com');
while (list ($line_num, $line) = each ($fcontents)) {
   echo "<b>Line $line_num:</b> " . htmlspecialchars ($line) .
"<br>\n";
}

?>
```

الدالة is_dir

عندما نريد ان ننشأ مجلد ويكون موجود يحدث خطأ يعطل البرنامج ، وقد نكون دائما بحاجة الى انشاء مجلدات في برنامجنا فلذلك يجب في البداية ان نعرف هل المجلد موجود وذلك عن طريق هذه الدالة فعندما نرسل للدالة اسم المجلد تفحص هل هذا الاسم موجود وهو هو مجلد فاذا كان كذلك تعود لنا بالقيمة ١ = (true) اما اذا لم يكن موجود او لم يكن مجلد فانه سوف يعود بالقيمة ٠ = (false) . اليك المثال :

```
echo is_dir ("mydir") ;
```

في هذا المثال نرسل الاسم mydir الى الدالة ونطبع نتيجة الاختبار .

الدالة mkdir

لانشاء المجلدات تحتاج الى هذه الدالة ، و عليك ان ترسل لها قيمتين الاولى اسم المجلد الذي تريد انشاءه والقيمة الثانية الترخيص الذي تود اعطاءه له ، وبالطبع اتوقع قبل استخدام هذه الدالة ان تستخدم الدالة السابقة لاختبار هل المجلد موجود لكي يكون الكود متكامل . واليك المثال :

```
echo mkdir  ("backup",0777) ;
```

في هذا المثال ننشأ المجلد backup ونعطيه ترخيص كامل للدخول والقراءة والكتابة .

الدالة rmdir

لحذف المجلد الذي أنشأناه في المثال السابق نستخدم هذه الدالة ، وكل ما علينا فعله هو ارسال اسم المجلد الى الدالة لتقوم بحذفه ، وبالطبع اتوقع قبل استخدام هذه الدالة ان تستخدم الدالة **is_dir** لاختبار هل المجلد موجود لكي يكون الكود متكامل . واليك المثال :

```
echo rmdir ("backup") ;
```

في هذا المثال نقوم بحذف المجلد backup .

الدالة rename

لتغيير اسم ملف نستخدم هذه الدالة ، ونحتاج هنا الى ارسال قيمتين ، القيمة الاولى اسم الملف والقيمة الثانية الاسم الجديد . واليك المثال :

```
echo rename ("backup.gif" , "backup1.gif") ;
```

في هذا المثال نغير اسم الملف backup.gif الى backup1.gif .

الدالة realpath

عن طريق هذه الدالة نتعرف على المسار الكامل والحقيقي للملف المحدد .

```
echo realpath ("backup.gif") ;
```

في هذا المثال نحصل على المسار الحقيقي للملف backup.gif .

الدالة tempnam

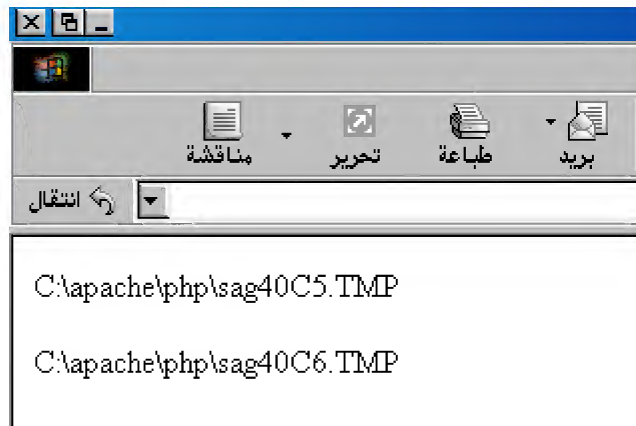
تقوم هذه الدالة بأنشاء ملف مؤقت وحيد أي اسمه غير موجود مسبقا .
وعندما نتعامل مع الكثير من البيانات نحتاج احيانا الى انشاء اكثر من ملف مؤقت ، ولكن من اهم الاخطاء الشائعة هي ان يكون اسم الملف موجود مسبقا ، مما يسبب خطأ في تنفيذ البرنامج ، لذلك نستخدم هذه الدالة .
الدالة تنشأ الملف في المجلد الذي ترسله لها ولكن اذا كان المجلد غير موجود فان الدالة آليا سوف تنشأ الملف في مجلد الملفات المؤقته في النظام وهو مثلا في الوندوز المجلد temp داخل مجلد الوندوز .

```
؟>

$tmpfname = tempnam ("", "sag");
echo $tmpfname ;
echo "</p>" ;
$tmpfname = tempnam ("", "sag");
echo $tmpfname ;

<؟
```

عند تنفيذ هذه الاوامر وحيث اننا سوف نستدعي هذه الدالة مرتين فسوف تقوم هذه الدالة بأنشاء ملفين مؤقتين اول ثلاث احرف منهما هي sag انظر الصورة :



امثلة مختلفة حول الملفات

لقد تعرفنا على الكثير من الدوال التي تتعامل مع الملفات لكي تساعدنا على التعامل و التحكم بالملفات والمجلدات ، والان سوف نأخذ بعض الامثلة للأستفادة من هذه الدوال :

المثال الاول : في هذا المثال سوف نتعلم كيف نفتح ملف موجود في نفس المجلد ، ثم نقوم بقراءته وطباعة الناتج ليراه المستخدم من خلال المتصفح .

الخطوات : ننشئ ملف ونسميه (test.dat) ونكتب في أي عبارة نود ، مثلا :

المبرمجون العرب
نخبة من الشباب العربي الطموح
يتميز بالطموح وحب المعرفة

الان نحفظ الملف .

وفي نفس المجلد ننشئ الملف (prog1.php) ونضع فيه الاوامر التالية :

```
<HTML dir=rtl>
<HEAD>
<TITLE>تجربة فتح وطباعة الملفات</TITLE>
</HEAD>
<BODY>
<?
        $myFile = fopen("test.dat","r");
        if(!($myFile))
        {
                print("لم اتمكن من فتح الملف");
                exit;
        }
        while(!feof($myFile))
        {
                $myLine = fgets($myFile, 255);
                print("$myLine <BR>\n");
        }
        fclose($myFile);
?>
</BODY>
</HTML>
```

الان نحفظ الملف وفي المتصفح نكتب عنوان البرنامج وليكن (http://localhost/myprog/prog1.php) وسوف تكون النتيجة التالي :



المبرمجون العرب

نخبة من الشباب العربي الطموح

يتميز بالطموح وحب المعرفة

المثال الثاني : في هذا المثال سوف نتعلم كيف نكتب الى ملف موجود ، واذا لم يكن موجود يقوم البرنامج بإنشائه . وسوف نكتب في هذا الملف تاريخ ووقت هذه الزيارة .

الخطوات : نحتاج فقط الى إنشاء ملف (prog2.php) وكتابة الاوامر التالية :

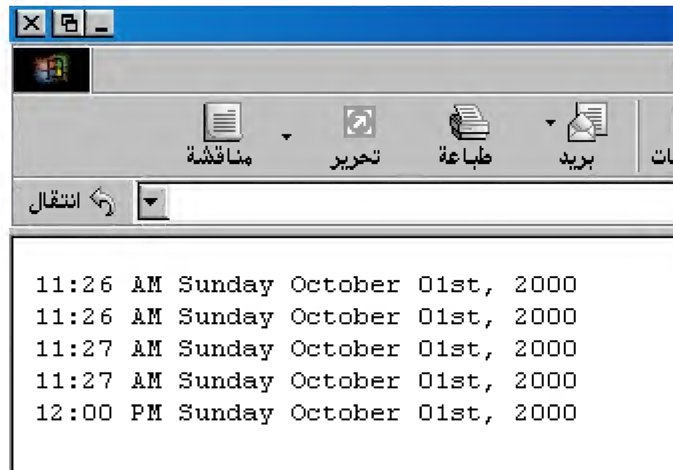
المفكرة - prog2.php

ملف تحرير بحث تعليمات

```
<HTML>
<HEAD>
<TITLE>الكتابة الى الملفات</TITLE>
</HEAD>
<BODY>
<?
        $fp = fopen("log.txt", "a");
        flock($fp, 2);
        fputs($fp, date("h:i A l F dS, Y\n"));
        flock($fp, 3);
        fclose($fp);

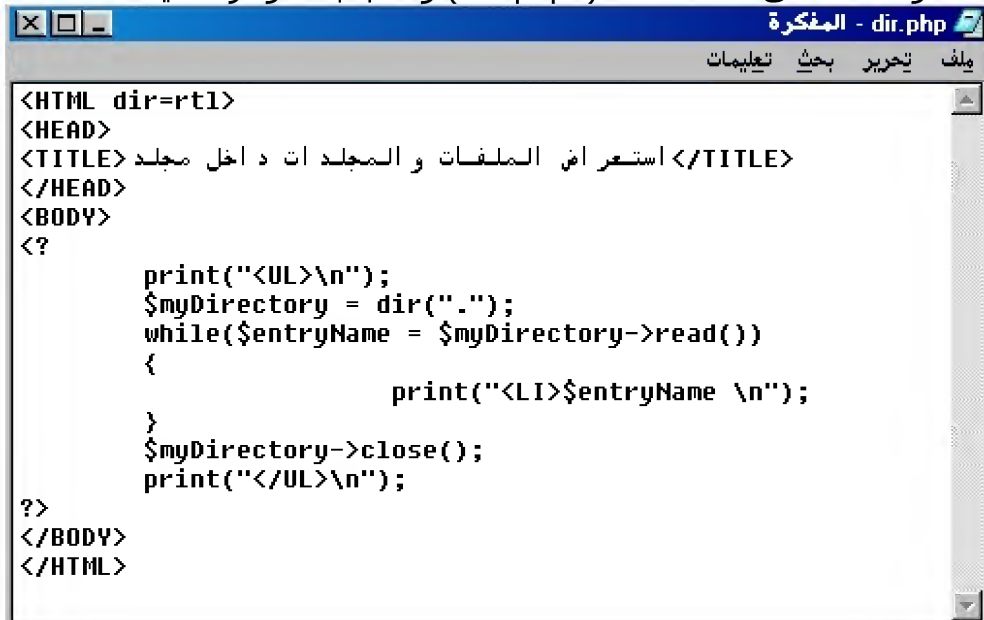
        print("<PRE>");
        readfile("log.txt");
        print("</PRE>\n");
?>
</BODY>
</HTML>
```

ثـم حفـظ الملـف . والان نسـتدعي الملـف بواسـطة كـتابة العـنوان التالـي (http://localhost/myprog/prog2.php) يجـب ان تنتـبه الـى ان هذا العنوان لدي وقد يكون مختلفاً لديك حسب المجلد الذي تنشأ فيه الملف .
وسوف تكون النتيجة كالتالي :



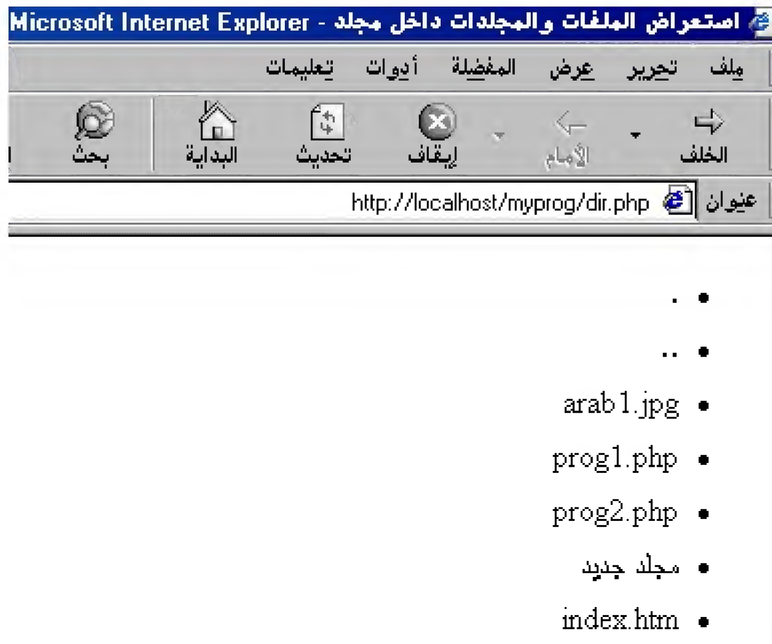
المثال الثالث : في هذا المثال سوف نتعرف على كيفية استعراض مجلد بحيث نطلع على جميع الملفات والمجلدات الموجوده داخله .

الخطوات : ننشئ ملف اسمه (dir.php) ونكتب به الاوامر التالية :




```
<HTML dir=rtl>
<HEAD>
<TITLE>استعراض الملفات و المجلدات داخل مجلد</TITLE>
</HEAD>
<BODY>
<?
        print("<UL>\n");
        $myDirectory = dir(".");
        while($entryName = $myDirectory->read())
        {
                        print("<LI>$entryName \n");
        }
        $myDirectory->close();
        print("</UL>\n");
?>
</BODY>
</HTML>
```

ثـم حفـظ الملـف . والان نسـتدعي الملـف بواسـطة كـتابة العـنوان التالـي (http://localhost/myprog/dir.php) يجـب ان تنتبه الى ان هذا العنوان لدي وقد يكون مختلفاً لديك حسب المجلد الذي تنشأ فيه الملف .
وسوف تكون النتيجة كالتالي :



بالطبع سوف نتعرض للتعامل مع الملفات عندما نأخذ الامثلة المتقدمة في فصل مخصص لمشاريع متقدمة في لغة php .

المصفوفات

تطرقنا للمصفوفات فيما سبق بشكل عابر ، والان سوف نتحدث عنها بالتفصيل وعن اهميتها وعن الدوال التي تتعامل مع المصفوفات .

تكمن اهمية المصفوفة في انها توفر الوقت والجهد بالاضافة الى انها تجعل البرنامج اصغر واسرع تنفيذا .

عموما المصفوفات يكون شكلها كما يلي :

$test = array ("a", "h", "a", "b", "b" , "a" , "d");

بحيث ان العنصر الاول في المصفوفة هو a والعنصر الثاني هو h وهكذا
ولطباعة قيمة العنصر الاول نكتب

Print $test[0] ;

لاحظ مايلي :
اسم المصفوفة : هو الاسم الذي نتعامل معه للحصول على المعلومات من المصفوفة .
رقم العنصر : لكل عنصر في المصفوفة رقم او اسم يدل عليه .

نستطيع كتابة المصفوفة ايضاً بأحد طريقتين انظر المثال التالي :

الطريقة الاولى :

```
$a["color"]      = "red";
$a["taste"]      = "sweet";
$a["shape"]      = "round";
$a["name"]       = "apple";
$a[3]            = 4;
```

الطريقة الثانية :

```
$a = array(
     "color" => "red",
     "taste" => "sweet",
     "shape" => "round",
     "name"  => "apple",
     3       => 4
);
```

نلاحظ ان البيانات في الطريقتين هما نفس البيانات ولكن اختلفت فقط طريقة كتابة المصفوفة .
وفي هذا المثال لدينا مصفوفة اسمها a ووضعنا بها عدة عناصر لكل عنصر قيمة . ولو اردنا معرفة قيمة العنصر المذاق (taste) لكتبنا التالي :

```
Print $a[taste] ;
```

اعتقد الان انه اتضحت طريقة عمل المصفوفات ولكن الا تلاحظ ان هذه المصفوفة بسيطة وتعود لنا فقط بمعلومات عنصر واحد .
ولذلك نسمي مثل هذه المصفوفات بالمصفوفات ذات البعد الواحد وهي تعمل لجلب صفات او مميزات عنصر واحد .
اذن ماذا لو كان لدينا عدة عناصر ونريد و نريد وضع مميزاتها في مصفوفة واحدة مثلا لدينا (البرتقال والتفاح والموز) ولكل واحد منها صفات مثل اللون والطعم والشكل فكيف نكتب هذه المصفوفة ذات الابعاد المتعددة ؟

كتابة المصفوفة ذات الابعاد المتعددة

ان المصفوفات قد تتشعب حتى تصبح معقدة جدا ، بل تصبح كشبكة العنكبوت لا تعرف اولا من آخرها ، ولكن لاتقلق ففي الحالات العادية لن تحتاج الى مصفوفات معقدة واغلب المصفوفات التى سوف تحتاجها ستكون ثنائية الابعاد فقط .
اذن دعنا نأخذ مثالا على مصفوفة ثنائية الابعاد وهو كما يلي :

array.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
$a = array(
     "apple"   => array(
          "color"   => "red",
          "taste"   => "sweet",
          "shape"   => "round"      ),
     "orange"   => array(
          "color"   => "orange",
          "taste"   => "tart",
          "shape"   => "round"      ),
     "banana"   => array(
          "color"   => "yellow",
          "taste"   => "paste-y",
          "shape"   => "banana-shaped"    )
);

?>
```

ولو اردنا معرفة مذاق التفاح فسوف نكتب التالي :

echo $a["apple"]["taste"];

لاحظ كتبنا اسم المصفوفة ثم اسم العنصر ثم الصفة .

بعد ان تعرفنا على طريقة عمل المصفوفات دعنا الان نتعرف على الدوال المتاحة في php التي تتعامل مع المصفوفات .

الدالة array

تعرفنا عليها سابقا لذلك سوف نشرح عملها هنا فقط ، وعملها هو انشاء مصفوفة . ويمكن بواسطتها انشاء مصفوفات متعددة الابعاد .

الدالة array_count_values

عندما نريد ان نتعرف على تكرار قيم في مصفوفة معينة فأننا نستخدم هذه الدالة وسوف تعود لنا بمصفوفة جديدة تحتوى على اسم العنصر وعدد مرات تكرره في المصفوفة .

مثال :

لنفترض لدينا المصفوفة التالية :

```
$test = array ("a", "h", "a", "b", "b" , "a" , "d");
```

نلاحظ انه في هذه المصفوفة ان الحرف a يتكرر ٣ مرات والحرف بـ يتكرر مرتين بينما الحرف d و h لا يتكرران .

اذن لمعرفة ذلك بواسطة هذه الدالة نكتب المثال التالي :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
$test = array ("a", "h", "a", "b", "b" , "a" , "d");
$count=array_count_values ($test);

        foreach($count as $key=>$value)
        {
                print("$key: $value <BR>\n");
        }
?>
```

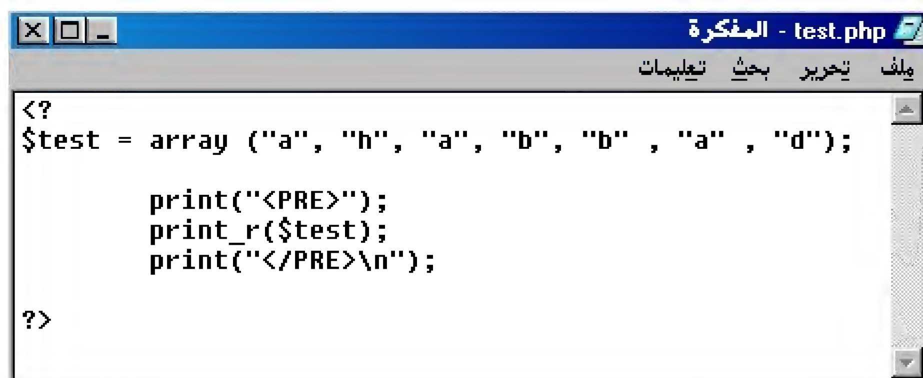
وهنا سوف يتم طباعة القيمة وعدد مرات تكررها في المصفوفة انظر الصورة التالية :



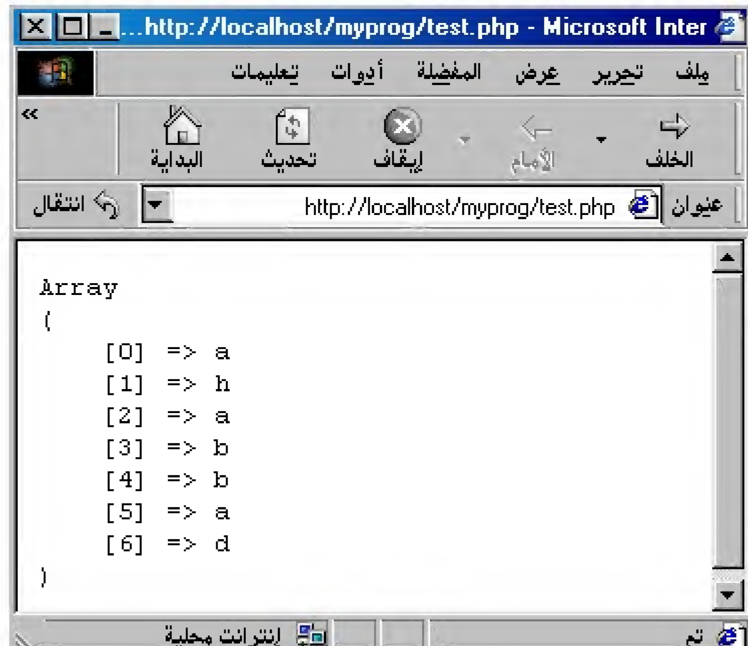
الدالة print_r

تقوم هذه الدالة بطباعة المصفوفة بشكل هرمي بحيث تتعرف على العناصر داخل هذه المصفوفة بكل سهولة .

مثال :



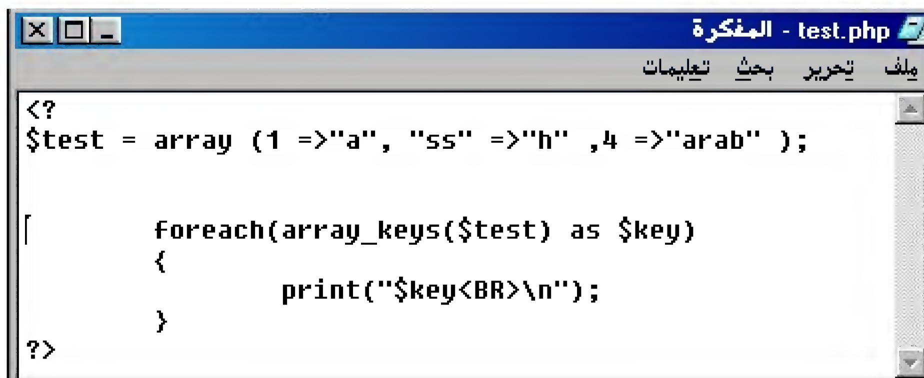
وعند تنفيذ هذا المثال سوف نحصل على التالي :



الدالة array_keys

عندما نتعامل مع المصفوفات من المهم جدا ان نعرف رقم او اسم العنصر لان هذا الرقم او الاسم هو من نستطيع من خلاله معرفة قيمة العنصر او الخاصية بالتحديد ، وحيث ان هناك الكثير من المصفوفات التي لاتعرف كم عنصر تعود به وماهي الارقام لهذه العناصر ، لذلك نستخدم هذه الدالة لتعود لنا فقط بأسم او رقم جميع العناصر الموجوده في المصفوفه .

مثال :



في هذا المثال لدينا ثلاث عناصر هي
الاول هو الرقم ١ وقيمته a
والثاني هو الاسم ss وقيمته h
والثالث هو الرقم ٤ وقيمته arab

ونحن هنا يهما المفاتيح الاساسية وهي (١ ، ss ، ٤) ولذلك استخدمنا الدالة **array_keys** لتعيد لنا اسماء هذه المفاتيح .

الدالة array_merge

عندما يكون لدينا مجموعة من المصفوفات ونرى انه من الافضل دمجها فلن نحتاج الى اعادة كتابة هذه المصفوفات من جديد ، بل نحتاج فقط الى ارسال اسمائها الى هذه الدالة لتقوم بكل العمليات بكل سهولة .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
$test1 = array ("a","b","c");
$test2 = array ("d","e","f");
$test=array_merge ($test1, $test2);

print("<PRE>") ;
print_r($test);
print("</PRE>\n") ;
?>
```

في هذا المثال لدينا مصفوفتان الاولى test1 والثانية test2 وقمنا بدمجهما في مصفوفة واحدة هي test وطلبنا من البرنامج طباعة عناصر المصفوفة الجديدة فيكون الناتج لدينا :

```
Array
(
    [0] => a
    [1] => b
    [2] => c
    [3] => d
    [4] => e
    [5] => f
)
```

الدالة array_pad

تقوم هذه الدالة بجعل قيم المصفوفة تساوي عددا معينا وتضع بدل القيم الناقصة القيمة الافتراضية التي ترسلها لها .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```php
<?
$test1 = array ("a","b","f");
$test=array_pad ($test1, 5,zzz);

print("<PRE>") ;
print_r($test);
print("</PRE>\n") ;
?>
```

تلاحظ اننا طلبنا ان يكون عدد عناصر المصفوفة ٥ فأذا كانت اقل من هذا العدد يتم اضافة القيم zzz في نهاية المصفوفة حتى يكتمل العدد ٥ .

بعد تنفيذ البرنامج نحصل على التالي :

```
Array
(
    [0] => a
    [1] => b
    [2] => f
    [3] => zzz
    [4] => zzz
)
```

تلاحظ ان تمت اضافة عنصرين رقم ٣ ورقم ٤ .

ملاحظة : لو اردنا ان نجعل القيم تضاف في بداية المصفوفة نكتب (-٥) بدلا من (٥) أي فقط نسبق الرقم بعلامة السالب .

الدالة array_pop

تقوم هذه الدالة بحذف آخر عنصر في المصفوفة .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
$test = array ("a","b","f");
$lastColor = array_pop($test);

        print("<PRE>");
        print_r($test);
        print("</PRE>\n");
?>
```

تلاحظ انه في المصفوفة test ثلاث عناصر وعند تنفيذ هذه الدالة عليها نحصل على التالي :

```
Array
(
    [0] => a
    [1] => b
)
```

كما تلاحظ اصبح في المصفوفة فقط عنصرين .

الدالة array_push

تقوم هذه الدالة بأضافة قيمة او اكثر الى نهاية الدالة .
مثال

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
$test = array ("a","b","f");
$lastColor = array_push ($test ,"d" ,"y");
        print("<PRE>");
        print_r($test);
        print("</PRE>\n");
?>
```

وعند تنفيذ هذا المثال نحصل على التالي :

```
Array
(
    [0] => a
    [1] => b
    [2] => f
    [3] => d
    [4] => y
)
```

الدالة array_reverse

تقوم هذه الدالة بعكس ترتيب عناصر المصفوفة من النهاية الى البداية ، بحيث يصبح العنصر الاول هو الاخير والعنصر الاخير هو الاول .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
$test1 = array ("a","b","f");
$test = array_reverse($test1);
        print("<PRE>");
        print_r($test);
        print("</PRE>\n");
?>
```

تلاحظ ترتيب عناصر المصفوفة اولا a ثم b واخيرا f وعند تنفيذ البرنامج :

```
Array
(
    [0] => f
    [1] => b
    [2] => a
)
```

تلاحظ ان الترتيب قد تم عكسه .

الدالة array_shift
تقوم هذه الدالة بحذف اول قيمة من المصفوفة .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
$test = array ("a","b","f");
$shift = array_shift($test);
        print("<PRE>");
        print_r($test);
        print("</PRE>\n");
?>
```

تلاحظ ان المصفوفة تحتوى على ثلاث عناصر وعند تنفيذ البرنامج :

```
Array
(
    [0] => b
    [1] => f
)
```

لقد تم حذف العنصر الاول من المصفوفة .

الدالة array_slice
تقوم هذه الدالة بنسخ مجموعة قيم من مصفوفة على شكل مصفوفة جديدة ، ويتم الحصول على القيم عن طريق تحديد بداية النسخ وعدد القيم التي تنسخ .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
$test1 = array ("a","b","f","d","g","p");
$test = array_slice($test1, 2,3);
        print("<PRE>");
        print_r($test);
        print("</PRE>\n");
?>
```

في هذا المثال نطلب من البرنامج نسخ القيم من المصفوفة ابتداء من القيمة رقم ٢ ويعود بعدد ٣ قيم وسوف تكون النتيجة :

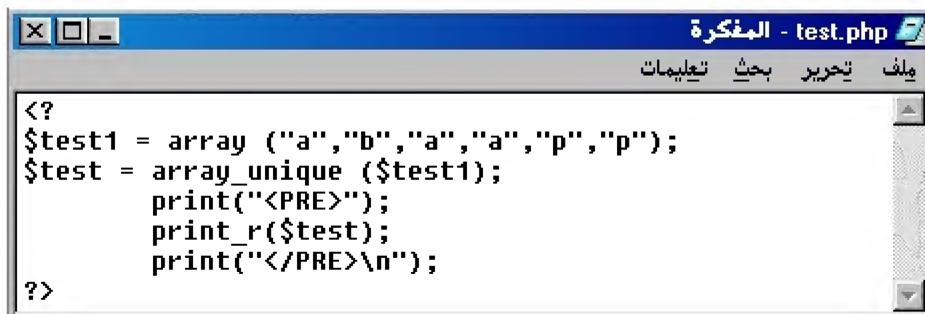
```
Array
(
    [0] => f
    [1] => d
    [2] => g
)
```

ملاحظة : عندما نريد الحصول على جميع القيم ابتداء من قيمة معينة فقط نرسل للدالة قيمة البداية .

الدالة array_unique

تقوم هذه الدالة بإرجاع القيم من مصفوفة بدون تكرار .

مثال :




```
<?
$test1 = array ("a","b","a","a","p","p");
$test = array_unique ($test1);
        print("<PRE>");
        print_r($test);
        print("</PRE>\n");
?>
```

وبعد تنفيذ البرنامج نحصل على :

```
Array
(
    [1] => b
    [3] => a
    [5] => p
)
```

تلاحظ انها اعادة لنا القيم بدون تكرار .

الدالة array_values

تعود هذه الدالة بجميع قيم المصفوفة في مصفوفة جديدة .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
$test1 = array ("a","b","a","a","p","p");
$test = array_values ($test1);
        for($n=0; $n < count($test); $n++)
        {
                print("$test[$n]<BR>\n");
        }
?>
```

وبعد تنفيذ البرنامج نحصل على التالي :

```
a
b
a
a
p
p
```

لاحظ اننا حصلنا على القيم دون ان نهتم بالمفتاح الاساسي للقيمة .

الدالة asort

تقوم هذه الدالة بترتيب القيم في المصفوفة من الاصغر الى الاكبر .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
$test = array ("a","b","a","a","p","p");
asort($test);
        print("<PRE>");
        print_r($test);
        print("</PRE>\n");

?>
```

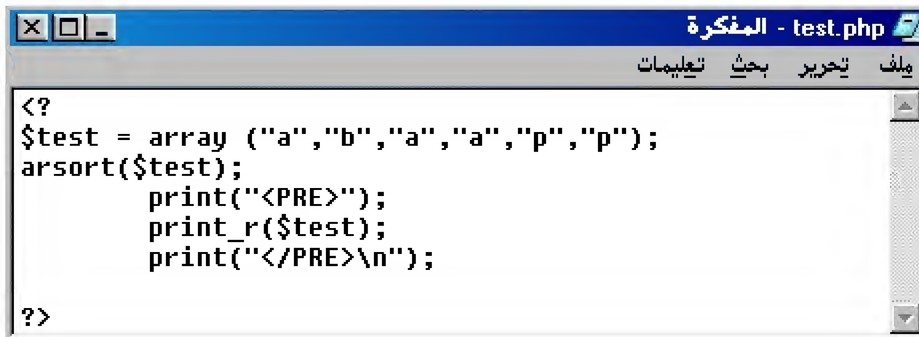
وعند تنفيذ هذا البرنامج نحصل على التالي :

```
Array
(
    [3] => a
    [2] => a
    [0] => a
    [1] => b
    [5] => p
    [4] => p
)
```

الدالة arsort

تقوم هذه الدالة بترتيب القيم في المصفوفة من الاعلى حتى الاصغر .

مثال :




```
<?
$test = array ("a","b","a","a","p","p");
arsort($test);
        print("<PRE>");
        print_r($test);
        print("</PRE>\n");

?>
```

وعند تنفيذ البرنامج نحصل على التالي :

```
Array
(
    [4] => p
    [5] => p
    [1] => b
    [3] => a
    [2] => a
    [0] => a
)
```

الدالة count

تقوم هذه الدالة بارجاع عدد العناصر في المصفوفة .

مثال :

المفكرة - test.php
ملف تحرير بحث تعليمات

```
<?
$test = array ("a","b","a");
        print count($test);

?>
```

وعند تنفيذ البرنامج نحصل على الرقم ٣ الذي يمثل عدد العناصر .

دوال الحركة لمؤشر قراءة المصفوفة

عندما نقوم بقراءة المصفوفة يقوم البرنامج بوضع مؤشر وهمي عند المتغير الذي نقرأه الان ومن خلال هذا المؤشر نستطيع التوجه للأمام او الخلف داخل المصفوفة واليك في الجدول التالي دوال الحركة وعمل كل منها :

الدالة	عملها
Reset	تصفير المؤشر ووضعه في بداية المصفوفة
next	التوجه للعنصر التالي
prev	التوجه الى العنصر السابق
Current	قيمة المتغير الحالي
End	وضع المؤشر عند اخر عنصر

مثال:

المفكرة - test.php
ملف تحرير بحث تعليمات

```
<?
$test = array ("a","b","a");
        for(reset($test); $value = current($test); next($test))
        {
                print("$value<BR>\n");
        }

?>
```

وعند تنفيذ البرنامج يتم طباعة القيم الموجودة في المصفوفة .

الدالة in_array

تعود بقيمة true اذا كانت القيمة موجودة داخل المصفوفة .

مثال :

```
<?
$test = array ("a","b","a");
if (in_array ("b", $test))
    print "found b";

?>
```

في هذا المثال اذا كانت القيمة موجوده سوف يتم طباعة الرسالة .

التعامل مع الارقام والدوال الرياضية

الان سوف نتعرف على الدوال التي تعطينا القدرة على التعامل مع الارقام والحصول منها على مخرجات محددة حسب متطلبات البرنامج .

الدالة bcadd

تتيح هذا الدالة لك جمع رقمين مهما كان نوعهما ، كما تتيح لك تحديد عدد الارقام بعد الفاصلة العشرية التي تدخل ضمن هذه العملية ، ونرسل لهذه الدالة ثلاث متغيرات الاول والثاني هما الارقام المطلوب جمعها بينما المتغير الثالث هو عدد الارقام بعد الفاصلة العشرية التي تدخل ضمن نطاق الجمع ، فلو ارسلنا لها صفر فسوف تهمل أي رقم خلف الفاصلة العشرية وتعود لنا بحاصل جمع الرقمين الصحيحين فقط .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
print(bcadd("1.59", "9.87654", 0));
print "<BR>" ;    // سطر جديد
print(bcadd("1.59", "9.87", 1));
print "<BR>" ;    // سطر جديد
print(bcadd("1.59", "9.87", 6));

?>
```

في هذا المثال وضعنا الحالات المحتملة لجمع رقمين وسوف تكون النتيجة كما يلي :

10
11.3
11.460000

تلاحظ ان النتيجة اختلفت فقط بتغيير قيمة المتغير الثالث الذي يتحكم بشكل المخرجات وطبيعتها .

الدالة bccomp

تقوم هذه الدالة بالمقارنة بين رقمين لتعود لك بنتيجة المقارنة فاذا كان الرقمان متساويان تعود بالقيمة صفر ، اما اذا كان الرقم الموجود باليسار اكبر من الرقم الموجود باليمين فتعود بالقيمة ١ ، واخيرا اذا كان الرقم الموجود باليمين اكبر من الرقم الموجود باليسار فسوف تعود بالقيمة -١ .
وهذه الدالة تتجاهل الارقام بعد الفاصلة العشرية الا اذا ارسلت لها قيمة ثالثة تحدد فيها عدد الارقام التي تضمن في عملية المقارنة .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
print(bccomp("1","1.11111"));
print "<BR>" ;    // سطر جديد
print(bccomp("1","1.111111111111",1));
print "<BR>" ;    // سطر جديد
print(bccomp("2","1.1111"));

?>
```

وسوف تكون النتيجة كما يلي :

```
0
-1
1
```

كما تلاحظ اعتبرت الدالة الرقمين الاولين متساويين لاننا طلبنا منها تجاهل الارقام بعد الفاصلة العشرية ، بينما الرقمين التاليين كانت الجهة اليمنى اكبر بعد ان طلبنا من الدالة ادخال رقم واحد بعد الفاصلة العشرية .
واخيرا كان الجانب الايسر اكبر لذلك عاد لنا بالقيمة ١ .

الدالة bcsqrt

تقوم هذه الدالة بأعادة الجذر التربيعي للرقم الذي سوف نرسله لها ، وايضا نستطيع ان نحدد عدد الارقام بعد الفاصلة العشرية التي نعرضها .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
print(bcsqrt("16"));
print "<BR>" ;    // سطر جديد
print(bcsqrt("17"));
print "<BR>" ;    // سطر جديد
print(bcsqrt("17",3));

?>
```

وسوف تكون لدينا النتيجة التالية :

4
4
4.123

تلاحظ ان جذر ١٦ و ١٧ كانت نتيجتهما متساوية عندما لم نطلب من الدالة ارجاع ارقام عشرية .
بينما العملية الاخيرة كان ناتجها مختلفا بحيث اشتمل على الارقام العشرية .

الدالة abs
تعود بالقيمة الحقيقية للرقم .

مثال :

```
print(abs(-13));
```

تعود بالرقم ١٣ .

الدالة max

تعود بأكبر قيمة من بين القيم المرسلة لها سواء كانت ارقام او نصوص .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
print(max(2, 5, 16));
print "<BR>" ;    // سطر جديد
print(max("arab", 15, 33));
print "<BR>" ;    // سطر جديد
print(max("arab", "sager"));

?>
```

وسوف تعود بالقيم التالية :

16
33
sager

الدالة Min
نفس طريقة عمل الدالة السابقة ولكنها تعود بأصغر قيمة من القيم المرسلة لها .

الدالة ceil

مهمة هذه الدالة تقريب الرقم الى اكبر رقم صحيح لاحق .

مثال:

```
print(ceil(13.01));
```

سوف نحصل على الرقم ١٤ .

الدالة log

الحصول على لوغارتم العدد .

مثال :

```
print(log(20.13));
```

الدالة sqrt

تعود بالجذر التربيعي للرقم المرسل لها .

مثال :

```
print(sqrt(9.9));
```

الدوال التي تتعامل مع المتغيرات

الدالة doubleval

بواسطة هذه الدالة نحول المتغير الى نوع الرقمي المضاعف من أي نوع آخر
.
مثال :

المفكرة - test.php

ملف تحرير بحث تعليمات

```
<?
$var = '200.234 Arab Programs';
$double_var = doubleval ($var);
print $double_var;

?>
```

وسوف تكون النتيجة طباعة الرقم 200.234 .

الدالة empty

عن طريق هذه الدالة نستطيع ان نعرف هل المتغير قد تم انشاءه وهل هو فارغ او يحمل القيمة صفر ، فهذه الدالة تعيد لنا القيمة false اذا كان المتغير موجود وليس فارغا ولا يساوي صفر .

مثال :

المفكرة - test.php

ملف تحرير بحث تعليمات

```
<?
$var = 0;
if (empty($var)) {
    print 'المتغير غير موجود او انه فارغ';
}
?>
```

الدالة gettype

بواسطة هذه الدالة نستطيع معرفو نوع المتغير .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
printf("%s <BR>\n", gettype(11));
printf("%s <BR>\n", gettype(7.3));
printf("%s <BR>\n", gettype("hello"));

?>
```

وسوف تكون النتيجة :

integer
double
string

الدالة intval

تحول هذه الدالة المتغير الى integer .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
$var="13.5cm" ;
$int_var = intval("13.5cm");
print $int_var

?>
```

الدالة is_array

تبين هذه الدالة هل المتغير عبارة عن مصفوفة ، تعود بالقيمة true اذا كانت المتغير عبارة عن مصفوفة .

مثال :

المفكرة - test.php

ملف تحرير بحث تعليمات

```
<?
$colors = array("red", "blue", "green");
  if(is_array($colors))
        {
          print("المتغير عبارة عن مصفوفة");
        }
?>
```

الدالة is_bool

تبين هذه الدالة هل المتغير من نوع Boolean .
اذا كان كذلك تعيد القيمة true .

مثال :

المفكرة - test.php

ملف تحرير بحث تعليمات

```
<?
$var = true ;
  if(is_bool($var))
        {
          print("المتغير من نوع boolean");
        }
?>
```

الدالة is_double
تبين هذه الدالة هل المتغير من نوع double .
اذا كان كذلك تعيد القيمة true .

مثال :

المفكرة - test.php

ملف تحرير بحث تعليمات

```
<?
$var = 200.234;
  if(is_double($var))
        {
          print("double المتغير من نوع");
        }
?>
```

الدالة is_int
تبين هذه الدالة هل المتغير من نوع integer .
اذا كان كذلك تعيد القيمة true .

مثال :

المفكرة - test.php

ملف تحرير بحث تعليمات

```
<?
$var = 200;
  if(is_int($var))
        {
          print("integer المتغير من نوع");
        }
?>
```

الدالة isset

تبين هذه الدالة هل تم تعريف المتغير ام لا .
اذا كان قد تم تعريف المتغير تعيد القيمة true .

مثال :

```
<?
$var = 200;
  if(isset($var))
        {
          print("المتغير موجود");
        }
?>
```

الدالة settype

تقوم هذه الدالة بإسناد نوع معين من البيانات الى متغير معين .

مثال :

```
settype($myValue, "integer");
```

الدالة unset

تقوم هذه الدالة بمسح المتغير من الذاكره .

مثال :

```
unset($var);
```

الدالة var_dump

تقوم هذه الدالة بتحديد نوع المتغير والبيانات الموجودة فيه .

مثال :

```
<?
    $a =100 ;
    var_dump ($a);
?>
```

دوال التعامل مع الصور

الدالة getimagesize

تقوم هذه الدالة بإعطائنا معلومات عن الصورة على شكل مصفوفة كما يلي :
العنصر الاول يعبر عن عرض الصورة .
العنصر الثاني يعبر عن ارتفاع الصورة .
العنصر الثالث يعبر عن نوع الصورة (١=GIF, 2=JPG, 3=PNG, 4=SWF)
العنصر الرابع : عرض وارتفاع الصورة مع النص وهذا العنصر غالبا يستخدم مع الوسم IMG لعرض الصور الذي يحتاج للعرض والارتفاع .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?
$image_file = "arab1.jpg";
$image_size = getimagesize($image_file);
print $image_size[0] ."<BR>";
print $image_size[1] ."<BR>";
print $image_size[2] ."<BR>";
print $image_size[3] ."<BR>";
print("<IMG SRC=\"$image_file\"
$image_size[3]><BR>\n");
?>
```

وسوف تكون النتيجة :

456
152
2
width="456" height="152"



الدالة imagecreate

انشاء صورة جديدة او مساحة عمل صورة جديدة . تحتاج هذه الدالة الى متغيرين يعبران عن عرض وارتفاع الصورة .
عند انشاء الصورة تعيد الدالة لنا مقبض وهذا المقبض مهم جدا وسوف نستخدمه في الدوال القادمة بشكل مستمر .

مثال :

```
$im = @ImageCreate (50, 100) ;
```

في هذا المثال عند انشاء الصورة او مساحة العمل سوف يعود لنا المقبض عن طريق المتغير $im .

الدالة imagecolorallocate

تعريف لون للصورة لاستخدامه في الرسم .
ونحتاج الى مقبض الصورة مع هذه الدالة .

مثال :

```
$colorMagenta = ImageColorAllocate ($im, 0, 0, 0);
```

كما تلاحظ ارسلنا لها مقبض النافذة بالاضافة الى متغيرات اللون .

الدالة imagearc

رسم قطع ناقص في مساحة العمل .
ونحتاج الى مقبض الصورة لكي نرسم عيها ولون الرسم .

مثال :

```
imagearc($im ,100, 50,100, 100,0, 360,$colorMagenta);
```

الدالة imagechar

رسم خط في مساحة العمل .
ايضا نحتاج الى مقبض الصورة لكي نرسم عيها ولون الرسم .
مثال :

```
imagechar($im , 1, 0, 0, "C", $colorMagenta);
```

الدالة imagefontwidth و imagefontheight

تحديد عرض وارتفاع الخطوط في مساحة العمل .

مثال :

```
$Width = imagefontwidth(20)   ;
$Height = imagefontheight(5) ;
```

الدالة imagejpeg

اخراج وحفظ الصورة في مساحة العمل الى المتصفح او ملف خارجي بصيغة jpg .

نحتاج الى مقبض الصورة .

مثال :

```
imagejpeg($im , "pic.jpg");
```

الدالة imageGIF

اخراج وحفظ الصورة في مساحة العمل الى المتصفح او ملف خارجي بصيغة gif .

نحتاج الى مقبض الصورة .

مثال :

```
imagegif($im , "pic.gif");
```

الدالة imagePNG

اخراج وحفظ الصورة في مساحة العمل الى المتصفح او ملف خارجي بصيغة png .

نحتاج الى مقبض الصورة .

مثال :

```
imagepng($im , "pic.png");
```

الدالة imageloadfont
تحميل خط معين .

مثال :

```
$myFont = imageloadfont("myFont") ;
```

الدالة imagestring
طباعة نص على مساحة العمل .
نحتاج الى مقبض الصورة و لون النص .

مثال :

```
imagestring($im , 4, 10, 10, "Arab Programmers",  $colorMagenta );
```

التعامل من سيرفر FTP

تستطيع من خلال php التعامل بكل احتراف مع سيرفر FTP والتحكم بالملفات والمجلدات .

الدالة ftp_connect

تقوم هذه الدالة باجراء اتصال مع سيرفر FTP فاذا تم الاتصال بنجاح تعود بمقبض لهذا الاتصال لنتمكن من استخدامه في بقية العمليات .

مثال :

```
$ftp = ftp_connect("4alarab.com")
```

الدالة ftp_login

تقوم هذه الدالة بتسجيل دخول المستخدم الى السيرفر .
وتحتاج هذه الدالة الى مقبض الاتصال والاسم وكلمة السر .
وتعود بالقيمة TRUE اذا تم الدخول بنجاح .

مثال :

```
$log=ftp_login($ftp, "user", "password") ;
```

الدالة ftp_pwd

تعود هذه الدالة بأسم المجلد الحالي الذي نتعامل معه .
تحتاج هذه الدالة الى مقبض الاتصال .

مثال :

```
$dir = ftp_pwd($ftp) ;
```

الدالة ftp_chdir

تقوم هذه الدالة بتغيير المجلد الحالي في السيرفر .
يحتاج الى مقبض الاتصال .

مثال :

```
$chdir = ftp_chdir($ftp, "www")
```

في هذا المثال نطلب الاتجاه الى المجلد www .

الدالة ftp_mkdir

تقوم هذه الدالة بانشاء مجلد جديد في المسار المحدد .
تحتاج الى مقبض الاتصال .

مثال :

```
$mkdir = ftp_mkdir($ftp, "sss")
```

الدالة ftp_rmdir

تقوم هذه الدالة بحذف مجلد معين في المسار المحدد .
تحتاج الى مقبض الاتصال .

مثال :

```
$rmdir = ftp_rmdir($ftp, "sss")
```

الدالة ftp_nlist

تقوم هذه الدالة يالعودة باسماء الملفات والمجلدات في المسار المحدد على شكل مصفوفة .
تحتاج الى مقبض الاتصال .

مثال :

```
$listf = ftp_nlist($ftp, ".")
```

الدالة ftp_get
تنزيل ملف من سيرفر FTP الى جهازك او موقعك .
تحتاج الى مقبض الاتصال .
وتحتاج الى نوع النقل هل هو FTP_ASCII او FTP_BINARY .

مثال :

```
ftp_get($ftp, "/tmp/data.bin", "/pub/data.bin", FTP_BINARY) ;
```

الدالة ftp_put
تقوم هذه الدالة بنقل الملف من جهازك الى سيرفر FTP .
تحتاج الى مقبض الاتصال .
وتحتاج الى نوع النقل هل هو FTP_ASCII او FTP_BINARY .

مثال :

```
ftp_put($ftp, "/tmp/data.bin", "/pub/data.bin", FTP_BINARY) ;
```

الدالة ftp_size
تعيد هذه الدالة حجم الملف المحدد .
تحتاج الى مقبض الاتصال .

مثال :

```
$size = ftp_size($ftp, “file.txt” ) ;
```

الدالة ftp_rename
تقوم هذه الدالة بتغيير الاسم للملف المحدد .
تحتاج الى مقبض الاتصال .

مثال :

```
$rename = ftp_rename($ftp, “file.txt” , “fileto.txt” ) ;
```

الدالة ftp_delete
تقوم هذه الدالة بحذف الملف المحدد .
تحتاج الى مقبض الاتصال .

مثال :

```
$delete = ftp_delete($ftp, “file.txt” ) ;
```

الدالة ftp_site
تقوم هذه الدالة بارسال الاوامر الى السيرفر .
تحتاج الى مقبض الاتصال .

مثال :

```
$send = ftp_site($ftp, “cd” ) ;
```

الدالة ftp_quit
تقوم هذه الدالة باغلاق الاتصال مع السيرفر .
تحتاج الى مقبض الاتصال .

مثال :

```
$close = ftp_quit($ftp) ;
```

التعامل مع قواعد البيانات MySQL

هل تعتقد ان كل برنامج بلا ثمن سوف يكون اقل جوده ؟
قد يكون ذلك مع كثير من البرامج ولكن محرك قواعد البيانات MySQL ليس كذلك ، فهو يعتبر من افضل محركات قواعد البيانات في العالم وخصوصا على الانترنت واصبح يضاهي اوراكل ومايكروسوفت اس كيو ال .
بل يتفوق عليهما لانه مجاني ومفتوح المصدر .

سوف نتعرف على هذا البرنامج بكل تفصيل في الخطوات التالية .

استخدام MySQL عن طريق موجه دوس

سوف نتعرف بهذه الخطوة على طريقة الاتصال بـ MySQL عن طريق موجه دوس وتنفيذ بعض الاوامر .
في البداية شغل محرك قواعد البيانات mysql .
شغل برنامج موجه دوس .
اذهب الى المسار (C:\apache\mysql\bin)
اكتب الامر (mysql) وسوف تشاهد النتيجة التالية :

```
MYSQL
C:\apache\mysql\bin>mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 3.23.32

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql>
```

الان نحن متصلون بمحرك قواعد البيانات .
الان نستطيع ان نستعرض :
قواعد البيانات المتوفرة لدينا .
الجداول الموجودة في كل قاعدة .
الاعمدة او الحقول الموجوده في الجداول .

لكي نشاهد قواعد البيانات المتوفرة لدينا نكتب الامر التالي :

Show databases;

وسوف تكون النتيجة شبيهه بالتالي :

```
MYSQL
Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql> show databases;
+----------+
| Database |
+----------+
| DLM      |
| bazar    |
| dbname   |
| linkat   |
| mysql    |
| new      |
| nuke     |
| nuketest |
| paFileDB |
| phpBB    |
| sager    |
| test     |
| test1    |
| vvb      |
+----------+
14 rows in set (0.00 sec)

mysql>
```

سوف تشاهد جميع قواعد البيانات الموجودة لديك .
الان لكي نتعامل مع قاعدة بيانات معينة يجب ان نخبر MySQL عنها ولنفترض اننا نريد ان نتعامل مع قاعدة البيانات المسماه mysql فيجب علينا كتابة الامر التالي :

Use mysql ;

الان يتوقع البرنامج ان أي اسفسار قادم له سوف يكون موجه حول هذه القاعدة .

الان لكي نشاهد الجداول الموجودة في هذه القاعدة نحتاج الى كتابة الامر التالي :

SHOW TABLES FROM mysql;

وسوف تشاهد النتيجة التالية :

```
mysql> show tables from mysql;
+-----------------+
| Tables_in_mysql |
+-----------------+
| columns_priv    |
| db              |
| host            |
| tables_priv     |
| user            |
+-----------------+
5 rows in set (0.11 sec)

mysql> _
```

كما تلاحظ لدينا خمس جداول في قاعدة البيانات هذه .

ولكي نتعرف على الحقول الموجودة ونوعها والشروط المفروضة على البيانات التي سوف تسجل بها نكتب الامر التالي :

SHOW COLUMNS FROM user;

لنشاهد التالي :

```
MYSQL
تلقائي

mysql> show columns from user;
+-----------------+---------------+------+-----+---------+-------+
| Field           | Type          | Null | Key | Default | Extra |
+-----------------+---------------+------+-----+---------+-------+
| Host            | char(60)      |      | PRI |         |       |
| User            | char(16)      |      | PRI |         |       |
| Password        | char(16)      |      |     |         |       |
| Select_priv     | enum('N','Y') |      |     | N       |       |
| Insert_priv     | enum('N','Y') |      |     | N       |       |
| Update_priv     | enum('N','Y') |      |     | N       |       |
| Delete_priv     | enum('N','Y') |      |     | N       |       |
| Create_priv     | enum('N','Y') |      |     | N       |       |
| Drop_priv       | enum('N','Y') |      |     | N       |       |
| Reload_priv     | enum('N','Y') |      |     | N       |       |
| Shutdown_priv   | enum('N','Y') |      |     | N       |       |
| Process_priv    | enum('N','Y') |      |     | N       |       |
| File_priv       | enum('N','Y') |      |     | N       |       |
| Grant_priv      | enum('N','Y') |      |     | N       |       |
| References_priv | enum('N','Y') |      |     | N       |       |
| Index_priv      | enum('N','Y') |      |     | N       |       |
| Alter_priv      | enum('N','Y') |      |     | N       |       |
+-----------------+---------------+------+-----+---------+-------+
17 rows in set (0.11 sec)

mysql>
```

الان لكي نتعرف على جميع البيانات الموجودة في الجدول user نستخدم الامر التالي :

Select * from user ;

عند تنفيذ هذا الامر سوف تلاحظ ان البيانات التي حصلت عليها كثيره ولا يمكن قراءتها ولكي نحصل على بيانات مرتبة وواضحه يجب ان نطلب بيانات اقل ، فمثلا يمكننا ان نطلب فقط الاسماء وكلمات السر الموجود في هذا الجدول وذلك بكتابة الامر التالي :

Select user , password from user ;

وعند تنفيذ هذا الامر نشاهد النتيجة التالية :

```
mysql> select user,password from user;
+------+----------+
| user | password |
+------+----------+
| root |          |
|      |          |
|      |          |
| root |          |
+------+----------+
4 rows in set (0.00 sec)

mysql> _
```

لاحظ اننا حصلنا على جميع البيانات الموجودة في الجدول وهي ٤ سجلات ولكن ماذا لو كنا نريد فقط الخاصة بالمستخدم root فاننا سوف نكتب الامر التالي :

Select user , password from user where user ='root' ;

الان سوف تشاهد النتيجة التالية :

```
mysql> select user , password from user where user='root' ;
+------+----------+
| user | password |
+------+----------+
| root |          |
| root |          |
+------+----------+
2 rows in set (0.11 sec)

mysql>
```

الان لنتعرف كيفية انشاء قاعدة بيانات بواسطة هذه الطريقة ، فمثلا لو اردنا ان ننشئ قاعدة البيانات data1 عن طريق كابة الامر التالي :

```
CREATE DATABASE mydata ;
```

سوف نشاهد رسالة تؤكد ان قاعدة البيانات قد تم انشائها .

لو اردنا مسح قاعدة البيانات هذه فسوف نكتب الامر التالي :

```
DROP DATABASE mydata ;
```

الان لو اردنا ان ننشئ جدول اسمه mytable فيه الحقول الرقم والاسم والدرجة سوف نكتب الكود التالي :
في البداية نخبر السيرفر علا اننا نستخدم القاعدة mudata بواسطة :

```
Use mydata ;
```

ثم نكتب الامر التالي :

```
CREATE TABLE mytable (no INT(5) NOT NULL ,
Name VARCHAR(20) NOT NULL ,
gride INT(3) NOT NULL ) ;
```

عند تنفيذ هذا الامر سوف تشاهد رسالة تخبرك بنجاح العملية .

ولو اردنا حذف الجدول السابق نكتب الامر التالي :

```
Drop table mytable ;
```

عند تنفيذ الامر السابق سوف تشاهد رسالة تخبر عن نجاح العملية .

الان لو اردنا اضافة بيانات الى الجدول نكتب الاوامر التالية :

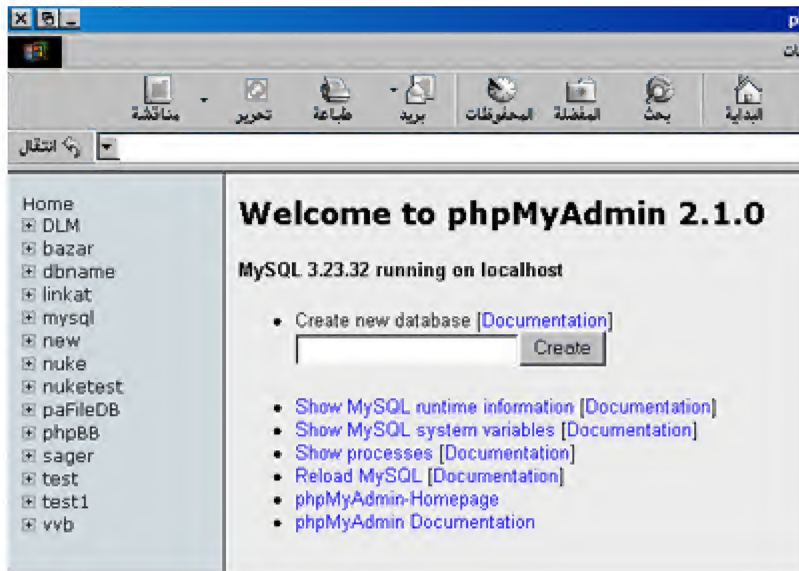
```
Insert into mytable (no,name,gride )
Value (1 ,’sager’ , 95 ) ;
```

عند تنفيذ الامر السابق سوف تشاهد رسالة تخبر عن نجاح العملية .

التحكم بقواعد البيانات بواسطة البرنامج phpMyAdmin

عـنـدما تـود الـتحكم بقواعـد الـبـيانات MySQL بشـكـل قوي وسـريع سـواء على الانتـرنـت او عـلـى جـهـازك الشـخـصي فأنت بحاجة الى البرنامج phpMyAdmin الذي يتيح لك تحكم متميز بقواعد البيانات .
وسـوف اقوم بشـرح طريقة اسـتخدام هذا البرنامج بكل تفصيل فيما يلي :

- الـبرنامج يكـون موجود مسبقا في أي مستضيف يدعم MySQL كما انه ينزل آلـيا مـع السـيرفر الشـخـصي لذلـك كل ماعليك لتشغيله كتابة العنوان التالي فـي متصـفحك (http://localhost/phpmyadmin) لتشـاهـد قواعـد البيانات الموجودة انظر الصورة :



الان لكي ننشأ قاعدة بيانات جديدة علينا كتابة اسـمها في الحقل Create new database ثم نضغط على الزر Create لنفترض اننا نريد انشـاء قاعدة اسـمها (mydata) .
الان تم انشـاء قاعدة البيانات ولكنها خالية انظر الشـكل التالي :

Home
DLM
bazar
dbname
linkat
mydata
mysql
new
nuke
nuketest
paFileDB
phpBB
sager
test
test1
vvb

Database mydata

Database mydata has been created.
No tables found in database.

- Print view
- Run SQL query/queries on database mydata [Documentation]:

or Location of the textfile: استعراض...
Go

- Query by Example
- View dump (schema) of database

Structure only — Add 'drop table' — Go
Structure and data — Send
Complete inserts

- Create new table on database mydata:

Name:
Fields: Go

ولكي ننشأ جدول جديد في قاعدة البيانات هذه نحتاج الى ادخال اسم الجدول في الخانة Create new table on database mydata: الموجود في اسفل الصفحة ثم ادخال عدد الحقول في هذا الجدول .
ليكن اسم الجدول mytable وعدد الحقول ٣ .
الان اضغط الزر Go . لتشاهد الشكل التالي :

Database mydata - table mytable

Field	Type	Length/Set	Attributes	Null
no	INT	5		not null
name	VARCHAR	30		not null
gride	INT	3		not null

Table comments:

Save

Default	Extra	Primary	Index	Unique

الا لنتعرف على ماذا تعني هذه المصطلحات :

Field : يطلب هنا اسم الحقل .
Type : نوع الحقل هل هو رقمي ام نص ام تاريخ (اختر من القائمة) .
Length/Set : طول البيانات الممكن ادخالها في الحقل .
Attributes : هل تود نسبة البيانات الى صفة معينة .
Null : هل يمكن تن يكون هذا الحقل فارغا (لا يحتوي على قيم) .
Default : القيمة الافتراضية لهذا الحقل .
Primary : هل هو مفتاح اساسي .
Index : هل هو مفهرس وهذا لسرعة البحث .
Unique : هل هو وحيد أي قيمته لا تتكرر .

بعد ان تدخل البيانات اضغط الزر Save لتشاهد الشاشة التالية :

Database mydata - table mytable

table mytable has been created.

SQL-query:
CREATE TABLE mytable (no INT (5) not null , name VARCHAR (30) not null , gride INT (3) not null)

Field	Type	Attributes	Null	Default	Extra	Action
no	int(5)		No	0		Change Drop Primary Index Unique
name	varchar(30)		No			Change Drop Primary Index Unique
gride	int(3)		No	0		Change Drop Primary Index Unique

- Print view
- Browse
- Select
- Insert

الان اصبح لدينا في الجدول mytable ثلاثة حقول ، ولكن هذه الحقول فارغة ولكي نضيف البيانات الى هذه الحقول نضغط على الكلمة insert الموجودة اسفل بيانات الجدول لنشاهد التالي :

Database mydata - table mytable

Field	Type	Function	Value
no	int(5)		1
name	varchar(30)		sager
gride	int(3)		95

Save

الان نضيف القيم الرقم والاسم والدرجة ثن نضغط الزر Save .

لقد تم اضافة ولكي نشاهد البيانات التي اضفناها نضغط على الكلمة Browse الموجودة ايضا اسفل بيانات الجدول لنشاهد التالي :

Database mydata - table mytable

SQL-query
SQL-query:
SELECT * FROM mytable LIMIT 0, 30

Begin << Previous < > Show 30 rows starting from >> End

no name gride
1 sager 95 Edit Delete

Begin << Previous < > Show 30 rows starting from >> End

Insert new row

كما تلاحظ يعرض البيانات الموجودة في الجدول ، ولو كانت البيانات كثيرة فسوف يعرض اول ٣٠ سطر ثم يمكنك التنقل الى المعلومات التالية بواسطة الازرة كما يلي :

الزر Begin : ينقلك الى اول ٣٠ سطر .
الزر Previous : ينقلك الى ٣٠ سطر سابقة .
الزر Show : يعرض ٣٠ سطر ابتداء مما وصلت اليه او من السطر الذي تحدده .
الزر End : يعرض آخر ٣٠ سطر .

الرقم ٣٠ تستطيع تغييره ولكنه وضع لتخفيف الضغط على السيرفر اثناء نقل البيانات .

كما تلاحظ ايضا لديك امام كل سطر الكلمتين :
Edit : لتغيير وتعديل البيانات في هذا السطر .
Delete : لحذف بيانات هذا السطر .
ايضا تشاهد في الاسفل الكلمة insert new row لاضافة سطر جديد .

تعليمات SQL

لغة SQL (Structured Query Language) هي لغة قياسية تستخدمها جميع قواعد البيانات للتعامل مع البيانات الموجودة في قاعدة البيانات سواء بالانشاء او الحذف او الاضافة او الاستعراض او التعديل .

وسوف نتحدث عن اهم تعليماتها بشكل مفصل هنا مع الامثلة التي سوف نطبقها على جدول الطلاب الذي اسميناه std وبه الحقول رقم الطالب no واسم الطالب name ودرجة الطالب gride وهو يحتوي على السجلات التالية :

no	name	lastname	gride
1	sager	mohammed	95
2	khaled	ahmad	93
3	sager	fahed	86
4	ahmad	mansor	68
5	sager	abrahem	67

التعليمة select

نستخدم التعليمة select لاسترجاع البيانات من جدول او جداول في قاعدة البيانات ، والصيغة الاساسة لهذه التعليمة هي :

```
Select  col1,col2,......  from   table1,table2,...... ;
```

كما تلاحظ فاننا في هذه الجملة كاننا نقول اننا نود استرجاع الحقول المحددة من الجداول المحددة .

مثال بسيط :
لو اردنا استرجاع اسم ورقم الطالب (name & no) من جدول الطلبة الذي اسمه (std) فاننا سوف نكتب الجملة التالية :

```
Select no,name from std ;
```

في هذه الجملة كاننا نقول : استرجع لنا الاسم والرقم من جدول الطلبة .
وفي هذه الحالة سوف يعيد لنا جميع اسماء وارقام الطلبة الموجودين في هذا الجدول .
لاحظ ايضا اننا فصلنا بين كل حقل واخر بعلامة الفاصلة .

وعند تنفيذ هذه الجملة نحصل على التالي :

```
mysql> select no,name from std ;
+----+--------+
| no | name   |
+----+--------+
|  1 | sager  |
|  2 | khaled |
|  3 | sager  |
|  4 | ahmad  |
|  5 | sager  |
+----+--------+
5 rows in set (0.00 sec)
```

ماذا لو اردنا استرجاع جميع الحقول في هذا الجدول ؟
علينا كتابة الجملة التالية :

Select * from std

بواسطة هذه الجملة سوف نسترجع جميع الحقول الموجود في الجدول وجميع السجلات الموجودة فيها .
عند تنفيذ هذه الجملة سوف نحصل على التالي :

```
mysql> select * from std ;
+----+--------+----------+-------+
| no | name   | lastname | gride |
+----+--------+----------+-------+
|  1 | sager  | mohammed |    95 |
|  2 | khaled | ahmad    |    93 |
|  3 | sager  | fahed    |    86 |
|  4 | ahmad  | mansor   |    68 |
|  5 | sager  | abrahem  |    67 |
+----+--------+----------+-------+
5 rows in set (0.00 sec)
```

التعليمة where

تلاحظ اننا في الجمل السابقة نحصل على جميع البيانات المتوفرة ، ولكن ليس هذا مانريده دائما .
ففي الغالب نحتاج الى درجة طالب واحد فقط او العشرة الاوائل فقط ، فتخيل عندما يأتي ولي امر الطالب ويطلب درجة ابنه فتعرض له بيانات جميع المدرسة امامه وتبحث فيها عن درجة هذا الطالب ، الا ترى ان هذا الامر محرج جدا ؟ بل ايضا لاداعي لوجود جهاز الكمبيوتر .

اذن دعنا نتعلم كيف نستخرج بيانات محددة من الجداول

لكي نستخرج بيانات محددة من الجدول نستخدم التعليمة where لكي نخبر التعليمة select بان تعيد لنا البيانات التي ينطبق عليها الشرط فقط .

الصيغة الاساسية للتعليمة where هي :

Select col1,col2,...... from table1,table2,...... where col1=??? ;

تلاحظ ان التعليمة select لم تتغير فقط اضفنا التعليمة where خلفها .

مثال :
لو اردنا الحصول على رقم واسم ودرجة الطالب (no,name,gride) من جدول الطلبة (std) للطالب الذي رقمه ٤ فاننا نستخدم الجملة التالية :

Select no,name,gride from std where no=4 ;

هنا نقول : نريد رقم واسم ادرجة الطالب من جدول الطلبة عندما يتحقق الشرط وهو ان يكون رقم هذا الطالب هو ٤ .
اذن سوف نشاهد الان نتائج هذا الطالب فقط كما يلي :

```
mysql> select no , name , gride from std where no=4 ;
+----+-------+-------+
| no | name  | gride |
+----+-------+-------+
|  4 | ahmad |    68 |
+----+-------+-------+
1 row in set (0.00 sec)
```

ولكن اذا كان ولي الامر لايعرف رقم ابنه ولكن كان اسمه sager فعندها نكتب الجملة التالية :

Select * from std where name='sager' ;

عند تنفيذ هذا الاستعلام لو كان لدينا في المدرسة اكثر من طالب بهذا الاسم فسوف نشاهد نتائجهم كما يلي :

```
mysql> select * from std where name='sager' ;
+----+-------+----------+-------+
| no | name  | lastname | gride |
+----+-------+----------+-------+
|  1 | sager | mohammed |    95 |
|  3 | sager | fahed    |    86 |
|  5 | sager | abrahem  |    67 |
+----+-------+----------+-------+
3 rows in set (0.00 sec)
```

في حالات كثيرة استخدام الجملة where لايقترن فقط بعلامة المساواة ولكننا نستطيع استخدام العلامات (=،<،>،=<،=>،<>) فمثلا لمشاهدة الطلبة الذين تكون درجاتهم اكبر من ٩٠ نكتب الجملة التالية :

Select * from std where grid > 90 ;

وعند تنفيذ هذه الجملة نحصل على النتيجة التالية :

```
mysql> select * from std where gride > 90 ;
+----+--------+----------+-------+
| no | name   | lastname | gride |
+----+--------+----------+-------+
|  1 | sager  | mohammed |    95 |
|  2 | khaled | ahmad    |    93 |
+----+--------+----------+-------+
2 rows in set (0.00 sec)
```

وايضا نستطيع استخدام علامات المقارنة التالية :

العلامة like

تعبر هذه العلامة عن مثل او يشبه او يحتوى على ، وهذه العلامة تتيح لنا مقارنة جزء من المعلومات .
فمثلا نستطيع ان نقول ان نطلب استرجاع السجلات التي تبدأ بالحرف s او السجلات التي تحتوي على حرف s عموما أي سواء كان في البداية ام النهاية ام في وسط الكلمة .
اعتقد انك عرفة قوة هذه العلامة ،

مثال ١ : لو اردنا ان نتعرف على الطلبة الذين تبدأ أسمائهم بحرف s سوف نكتب الجملة التالية :

Select * from std where name like 's%' ;

عند تنفيذ هذ الجملة تشاهد التالي :

```
mysql> select * from std where name like 's%' ;
+----+-------+----------+-------+
| no | name  | lastname | gride |
+----+-------+----------+-------+
|  1 | sager | mohammed |    95 |
|  3 | sager | fahed    |    86 |
|  5 | sager | abrahem  |    67 |
+----+-------+----------+-------+
3 rows in set (0.06 sec)
```

مثال ٢ : لو اردنا التعرف على الطلبة الذين يوجد من ضمن احرف اسمائهم الحرف h في أي مكان من الاسم نكتب الجملة التالية :

Select * from std where name like '%h%' ;

وعند تنفيذ هذه الجملة نشاهد التالي :

```
mysql> select * from std where name like '%h%' ;
+----+--------+----------+-------+
| no | name   | lastname | gride |
+----+--------+----------+-------+
|  2 | khaled | ahmad    |    93 |
|  4 | ahmad  | mansor   |    68 |
+----+--------+----------+-------+
2 rows in set (0.05 sec)
```

العلامة IN

يمكن تعريف هذه العلامة على انها الشرط (توفر احد هذه القيم) أي ان الشرط يتحقق اذا كان احد هذه القيم صحيح .

مثال :
لو اردنا درجات الطلبة الذين ارقامهم هي ١ و ٣ و ٤ فاننا سوف نكتب الجملة التالية :

Select * from std where no in (1,3,4) ;

وعند تنفيذ هذه الجملة نحصل على التالي :

```
mysql> select * from std where no in (1,3,4) ;
+----+-------+----------+-------+
| no | name  | lastname | gride |
+----+-------+----------+-------+
|  1 | sager | mohammed |    95 |
|  3 | sager | fahed    |    86 |
|  4 | ahmad | mansor   |    68 |
+----+-------+----------+-------+
3 rows in set (0.06 sec)
```

العلامة Between

أي نحتاج الى القيم الواقعة بين هاتين القيمتين وتستخدم عادة في البحث بين تاريخين معينين .

مثال :

عندما نحتاج الى الطلبة الذين درجاتهم بين ٨٠ و ٩٠ نكتب الجملة التالية :

Select * from std where gride between 80 and 90 ;

وعند تنفيذ هذه الجملة تحصل على النتيجة التالية :

```
mysql> select * from std where gride between 80 and 90 ;
+----+-------+----------+-------+
| no | name  | lastname | gride |
+----+-------+----------+-------+
|  3 | sager | fahed    |    86 |
+----+-------+----------+-------+
1 row in set (0.05 sec)
```

دمج الشروط

نستطيع كتابة اكثر من شرط على نفس الحقل او حقول متعددة ، ولكي نفعل ذلك نستخدم الكلمتين التاليتين :

And : يجب تحقق الشرطين .

Or : تحقق احد الشرطين .

مثال :

لو اردنا ان نعرف الطلبة الذين ارقامهم اكبر من ٢ ودرجاتهم اكبر من ٧٠ نكتب الجملة التالية :

Select * from std where no > 2 and gride > 70 ;

وعند تنفيذ هذه الجملة نحصل على النتيجة التالية :

```
mysql> select * from std where no > 2 and gride > 70 ;
+----+-------+----------+-------+
| no | name  | lastname | gride |
+----+-------+----------+-------+
|  3 | sager | fahed    |    86 |
+----+-------+----------+-------+
1 row in set (0.00 sec)
```

بينما لو استخدمنا الكلمة or مع هذه الجملة السابقة كما يلي :

Select * from std where no > 2 or gride > 70 ;

وعند تنفيذ هذه الجملة نحصل على النتيجة التالية :

```
mysql> select * from std where no > 2 or gride > 70 ;
+----+--------+----------+-------+
| no | name   | lastname | gride |
+----+--------+----------+-------+
|  1 | sager  | mohammed |    95 |
|  2 | khaled | ahmad    |    93 |
|  3 | sager  | fahed    |    86 |
|  4 | ahmad  | mansor   |    68 |
|  5 | sager  | abrahem  |    67 |
+----+--------+----------+-------+
5 rows in set (0.10 sec)
```

تلاحظ اننا عند تطبيق المعيار OR حصلنا على جميع البيانات لان الشرط يتحقق بتحقق احدى النتائج أي ان يكون الرقم اكبر من ٢ او تكون الدرجة اكبر من ٧٠ .

التعليمة order by
نحتاج في كثير من الاحيان الى ترتيب البيانات حسب صف معين ، ولذلك نستخدم هذه التعليمة .
وتأتي مع هذه التعليمة الكلمتان :
Asc : ترتيب الحقل من الاصغر الى الأكبر (هو الافتراضي) .
Desc : ترتيب الحقل من الاكبر الى الاصغر .

مثال : للحصول على البيانات مرتبه حسب الدرجة من الاصغر الى الاكبر نكتب الجملة التالية :

Select * from std order by gride asc ;

عند تنفيذ هذه الجملة نحصل على التالي :

```
mysql> select * from std order by gride asc ;
+----+--------+----------+-------+
| no | name   | lastname | gride |
+----+--------+----------+-------+
|  5 | sager  | abrahem  |    67 |
|  4 | ahmad  | mansor   |    68 |
|  3 | sager  | fahed    |    86 |
|  2 | khaled | ahmad    |    93 |
|  1 | sager  | mohammed |    95 |
+----+--------+----------+-------+
5 rows in set (0.05 sec)
```

بينما لو اردنا الحصول على المعلومات السابقة مرتبة حسب الدرجات من الاكبر الى الاصغر نكتب الجملة التالية :

Select * from std order by gride desc ;

عند تنفيذ هذه الجملة نحصل على التالي :

```
mysql> select * from std order by gride desc ;
+----+--------+----------+-------+
| no | name   | lastname | gride |
+----+--------+----------+-------+
|  1 | sager  | mohammed |    95 |
|  2 | khaled | ahmad    |    93 |
|  3 | sager  | fahed    |    86 |
|  4 | ahmad  | mansor   |    68 |
|  5 | sager  | abrahem  |    67 |
+----+--------+----------+-------+
5 rows in set (0.00 sec)
```

التعليمة limit

هـذه التعلـيمة تحـدد عـدد السجلات الاقصى الذي نحصل عليه ، تخيل عندما تـتعامل مـع شــبكة ومـع قاعدة بيانات ضخمة فلا اعتقد في الغالب انك سـوف تقـرأ او تـتعامل مـع جمـيع البـيانات فـي نفـس اللحظـة ولذلـك مـن الافضـل اسـترجاع البـيانات مجـزئة مما يخفـف الضـغط علـى الشـبكة ويتـيح للآخرين الاستفادة منها .

مثال : عند الحاجة الى استرجاع اول ثلاث سجلات فقط نكتب الجملة التالية :

Select * from std limit 3 ;

وعند تنفيذ هذه الجملة نحصل على التالي :

```
mysql> select * from std limit 3 ;
+----+--------+----------+-------+
| no | name   | lastname | gride |
+----+--------+----------+-------+
|  1 | sager  | mohammed |    95 |
|  2 | khaled | ahmad    |    93 |
|  3 | sager  | fahed    |    86 |
+----+--------+----------+-------+
3 rows in set (0.00 sec)
```

وعندما نريد استعراض السجلين المتبقيين نكتب الجملة التالية :

Select * from std limit 3,2 ;

وعند تنفيذ هذه الجملة نحصل على التالي :

```
mysql> select * from std limit 3,2 ;
+----+-------+----------+-------+
| no | name  | lastname | gride |
+----+-------+----------+-------+
|  4 | ahmad | mansor   |    68 |
|  5 | sager | abrahem  |    67 |
+----+-------+----------+-------+
2 rows in set (0.00 sec)
```

كما تلاحظ وضعنا خلف التعليمة limit رقمين يعبر الاول عن الرقم الذي نبدأ بعده استيراد البيانات والرقم الاخر هو عدد البيانات التي نحضرها .

التعليمة insert

عندما تود اضافة بيانات الى الجدول تحتاج الى هذه التعليمة .

مثال : لاضافة بيانات طالب جديد اسمه الاول yosef واسمه الاخير rakan ودرجته ٩٢ وسوف نعطيه الرقم ٦ سوف نكتب الجملة التالية :

Insert into std (no , name , lastname , gride)
Values (6 , 'yosef ' , 'rakan' , 92) ;

وعند تنفيذ هذه الجملة نحصل على رسالة تأكد انه تم اضافة سجل واحد الى الجدول ، ولكي نتأكد من ذلك نستخدم الجملة التالية :

Select * from std ;

لنشاهد التالي :

```
mysql> select * from std ;
+----+--------+----------+-------+
| no | name   | lastname | gride |
+----+--------+----------+-------+
|  1 | sager  | mohammed |    95 |
|  2 | khaled | ahmad    |    93 |
|  3 | sager  | fahed    |    86 |
|  4 | ahmad  | mansor   |    68 |
|  5 | sager  | abrahem  |    67 |
|  6 | yosef  | rakan    |    92 |
+----+--------+----------+-------+
6 rows in set (0.00 sec)
```

ايضا نستطيع كتابة نفس الجملة السابقة بالطريقة التالية :

Insert into std Values (6 , 'yosef ' , 'rakan' , 92) ;

ولكن يجب ان نكتب الحقول بالترتيب حسب ترتيبها في الجدول .

التعليمة update

نستخدم هذه التعليمة لتعديل بيانات محدده في الجدول .

مثال : لو ان الطالب رقم ٣ اوضح ان اسمه الاخير تغير من mansor الى sultan وطلب منا تعديل ذلك فسوف نستخدم الجملة التالية :

```
Update std set lastname = 'sultan' where no = 3 ;
```

وعند تنفيذ هذه الجملة نحصل على رسالة تبين ان العملية تمت بنجاح .

ولنتأكد من ذلك نكتب الجملة التالية :

```
Select * from std ;
```

لنشاهد التالي :

```
mysql> select * from std ;
+----+--------+----------+-------+
| no | name   | lastname | gride |
+----+--------+----------+-------+
|  1 | sager  | mohammed |    95 |
|  2 | khaled | ahmad    |    93 |
|  3 | sager  | sultan   |    86 |
|  4 | ahmad  | mansor   |    68 |
|  5 | sager  | abrahem  |    67 |
|  6 | yosef  | rakan    |    92 |
+----+--------+----------+-------+
6 rows in set (0.00 sec)
```

لاحظ اننا استخدمنا مع التعليمة update التعليمة where لنحدد السجلات التي سوف يحدث لها التغيير فلو كتبنا هذه الجملة :

```
Update std set lastname = 'sultan'   ;
```

هذه الجملة سوف تغير الاسم الاخير لجميع السجلات الى الاسم الجديد .

التعليمة delete

عندما نحتاج لحذف سجل من الجدول نستخدم هذه التعليمة .

مثال : اذا انسحب الطالب رقم ٤ من المدرسة ولم يعد هناك حاجة للاحتفاظ ببياناته ، وطلب منا حذف البيانات فسوف نستخدم الجملة التالية :

Delete from std where no = 4 ;

وعند تنفيذ هذه الجملة نحصل على رسالة تبين ان العملية تمت بنجاح .

ولنتأكد من ذلك نكتب الجملة التالية :

Select * from std ;

لنشاهد التالي :

```
mysql> select * from std ;
+----+--------+----------+-------+
| no | name   | lastname | gride |
+----+--------+----------+-------+
|  1 | sager  | mohammed |    95 |
|  2 | khaled | ahmad    |    93 |
|  3 | sager  | sultan   |    86 |
|  5 | sager  | abrahem  |    67 |
|  6 | yosef  | rakan    |    92 |
+----+--------+----------+-------+
5 rows in set (0.06 sec)
```

لاحظ اننا استخدمنا مع التعليمة delete التعليمة where لنحدد السجلات التي سوف يتم حذفها فلو كتبنا هذه الجملة :

Delete from std ;

هذه الجملة سوف تحذف جميع السجلات في الجدول .

التعليمة Count

للحصول على عدد السجلات من حقل معين .

مثال : لنتعرف على عدد الطلبة الموجودين لدينا في قاعدة البيانات نكتب الجملة التالية :

Select count(*) from std ;

وعند تنفيذ هذه الجملة نشاهد التالي :

```
mysql> select count(*) from std ;
+----------+
| count(*) |
+----------+
|        5 |
+----------+
1 row in set (0.06 sec)
```

التعليمة MAX

للحصول عل اكبر قيمة من السجلات .

مثال : للحصول على اكبر درجة حصل عليها طالب نكتب الجملة التالية :

Select max(gride) from std ;

وعند تنفيذ هذه الجملة نحصل على التالي :

```
mysql> select max(gride) from std ;
+------------+
| max(gride) |
+------------+
|         95 |
+------------+
1 row in set (0.27 sec)
```

التعليمة MIN
للحصول على اقل قيمة في السجلات .

مثال : للحصول على اقل درجة حصل عليها طالب نكتب الجملة التالية :

Select min(gride) from std ;

وعند تنفيذ هذه الجملة نحصل على التالي :

```
mysql> select min(gride) from std ;
+------------+
| min(gride) |
+------------+
|         67 |
+------------+
1 row in set (0.06 sec)
```

التعليمة AVG
للحصول على معدل السجلات .

مثال : للحصول على معدل الدرجات التي حصل عليها طالب نكتب الجملة التالية :

Select avg(gride) from std ;

وعند تنفيذ هذه الجملة نحصل على التالي :

```
mysql> select avg(gride) from std ;
+------------+
| avg(gride) |
+------------+
|    86.6000 |
+------------+
1 row in set (0.06 sec)
```

التعليمة SUM

للحصول على مجموع بيانات .

مثال : لنتعرف على مجموع الدرجات التي حصل عليها جميع الطلبة نكتب الجملة التالية :

Select sum(gride) from std ;

وعند تنفيذ هذه الجملة نحصل على التالي :

```
mysql> select sum(gride) from std ;
+------------+
| sum(gride) |
+------------+
|        433 |
+------------+
1 row in set (0.00 sec)
```

PHP & MySQ

الان سوف نتعرف على كيفية الاتصال والتحكم بـ MySQL بواسطة PHP لان المستخدم الذي يزور موقعك لن يكون لديه صلاحية استخدام البرامج او الدخول واستخدام قاعدة البيانات الموجودة في موقعك مباشرة ، ولكنه سوف يجد الشاشات التي تحددها له بحسب الصلاحيات المحددة ايضا .

لكي تستفيد من قاعدة البيانات في MySQL عليك باتباع خطوتين مهمتين هما :
الخطوة الاولى : اجراء عملية الاتصال بقاعدة البيانات ونجاح هذا الاتصال ولكي نقوم بهذه العملية نستخدم الدالة التالية :

الدالة mysql_connect

هذه الدالة تستخدم لإجراء الاتصال بقاعدة البيانات وهي تحتاج الى ثلاث متغيرات :
المتغير الاول : اسم السيرفر .
المتغير الثاني : اسم المستخدم .
المتغير الثالث : كلمة السر .

وتعود هذه الدالة بمقبض الاتصال في حالة نجاح الاتصال او رسالة تعبر عن الخطأ في حالة فشل الاتصال .

مثال :

test.php - المفكرة
ملف تحرير بحث تعليمات

```
<?php
    $link = mysql_connect ("localhost", "root", "")
        or die ("فشل الاتصال");
    print ("تم الاتصال");
    mysql_close ($link);
?>
```

في هذا المثال عند حدوث الاتصال سوف يتم طباعة العبارة تم الاتصال ، ولكن عند فشل الاتصال تطبع العبارة فشل الاتصال وتلاحظ اننا قمنا بانهاء الاتصال بواسطة الدالة mysql_close .

الخطـوة الثانـية : تحديد قاعدة البيانات التي تود التعامل معها من خلال الدالة التالية :

الدالة mysql_select_db

هـذه الدالـة تحـدد قاعدة البيانات التي نتعامل معها حاليا ، وتحتاج هذه الدالة الى متغيرين هما :
المتغير الاول : اسـم قاعدة البيانات التي نود التعامل معها .
المتغير الثاني : مقبض الاتصال الذي اجريناه .

و تعـود هـذه الدالـة بالقـيمة true في حالة نجاحها او false في حالة حدوث أي خطأ .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?php
    $link = mysql_connect ("localhost", "root", "")
        or die ("فشل الاتصال");
    print ("تم الاتصال");
    mysql_select_db("school", $link);

?>
```

الان بعـد نجـاح الاتصـال وتحديـد قـاعدة البـيانات التـي سـوف تـتعامل معهـا تسـتطيع الحصـول او الـتحكم بجمـيع الجداول الموجودة بقاعدة البيانات ولكي نفعل ذلك يوفر PHP لنا عدة دوال سوف نشرحها فيما يلي .

الدالة mysql_affected_rows

تعـود لـنا هـذه الدالة بعدد السجلات التي تأثرت من احد العمليات (delete , insert , update) .
وتحتاج هذه الدالة الى متغير واحد وهو مقبض الاتصال .

ملاحظـة : عـند اسـتخدام التعليمة delete بدون استخدام where فسوف يتم حذف جميع السجلات في الجدول ولكن الدالة سوف تعيد لنا القيمة صفر في هذه الحالة .

الدالة mysql_num_rows

تعـود لـنا هذه الدالة بعدد السجلات التي حصلنا عليها بعد استـخدام التعليمة select فقط .
وتحتاج هذه الدالة الى متغير واحد وهو مقبض الاتصال .

ملاحظة : لاتستـخدم هذه الدالة مع التعليمات (delete , insert , update) .

الدالة mysql_query

تـنفذ هـذه الدالـة جمـل الاستعلام على قواعد البيانات ، وتعود بالقيم الناتجة عن عملية الاستعلام على شكل مقبض لعملية الاستعلام .

مثال :

```
$result = mysql_query("SELECT * frm std");
```

تلاحـظ ان القـيم المســتخرجة وضعت في المتغير result ولكي نستفيد من هذه القيم نستخدم الدالة التالية :

الدالة mysql_fetch_row

تضـع هـذه الدالـة القـيم المســترجعة في مصفوفة لكي نتعامل معها بسـهولة وتحتاج فقط الى مقبض عملية الاستعلام .

مثال :

```
$row = mysql_fetch_array($result)
```

الان لديـنا مصـفوفة نسـتطيع التعامل معها بكل سـهولة (سـوف نرى ذلك في عدة امثله حول التعامل مع قواعد البيانات) .

الدالة mysql_create_db

تستخدم هذه الدالة لانشاء قاعدة بيانات جديدة ، وتحتاج فقط الى اسم هذه القاعدة .
وتعود بالقيمة **true** في حالة نجاح العملية .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?php
    $link = mysql_connect ("localhost", "root", "")
        or die ("فشل الاتصال");

    if (mysql_create_db ("mydatabase")) {
        print ("تم انشاء قاعدة البيانات");
    } else {
        print (" حدث خطأ ");
    }

?>
```

الدالة mysql_drop_db

تســتخدم هذه الدالة لحذف قاعدة البيانات وجميع الجداول الموجوده فيها ، وتحتاج فقط الى اسم قاعدة البيانات المطلوب حذفها .
وتعود بالقيمة **true** في حالة نجاح العملية .

مثال :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<?php
    $link = mysql_connect ("localhost", "root", "")
        or die ("فشل الاتصال");

    if (mysql_drop_db ("mydatabase")) {
        print ("تم حذف قاعدة البيانات");
    } else {
        print (" حدث خطأ ");
    }

?>
```

الدالة mysql_list_dbs

بواسطة هذه الدالة نستطيع معرفة قواعد البيانات المتوفرة لدينا ، وتحتاج فقط الى مقبض الاتصال .

مثال :

$Result = mysql_list_dbs($Link);

تلاحظ ان القيم المستخرجة وضعت في المتغير **result** ولكي نستفيد من هذه القيم نستخدم الدالة **mysql_fetch_row** التي تعرفنا عليها مسبقا .

الدالة mysql_list_tables

بواسطة هذه الدالة نستطيع معرفة الجداول الموجودة في قاعدة البيانات ، وتحتاج هذه الدالة الى متغير واحد هو اسم قاعدة البيانات .

مثال :

$Result = mysql_list_tables("mydatabase") ;

تلاحظ ان القيم المستخرجة وضعت في المتغير **result** ولكي نستفيد من هذه القيم نستخدم الدالة **mysql_fetch_row** التي تعرفنا عليها مسبقا .

الدالة mysql_list_fields

بواســطة هــذه الدالة نستطيع معرفة الحقول الموجودة في جدول معين ، وتحتاج هذه الدالة لمتغيرين :
المتغير الاول : اسم قاعدة البيانات .
المتغير الثاني : اسم الجدول .
وســوف تعــود هذه الدالة بمقبض للعملية لنستخدمه في معرفة معلومات حقول الجدول سواء الاسم او النوع او الطول او الشرط بواسطة الدوال التالية :

الدالة **mysql_field_name** : لمعرفة اسم الحقل .
الدالة **mysql_field_type** : لمعرفة نوع الحقل .
الدالة **mysql_field_len** : لمعرفة طول الحقل .
الدالة **mysql_field_flags** : الشروط على الحقل .

وسوف نأخذ مثال مفصل حول هذه الدالة .

امثلة على التعامل مع قواعد البيانات

المثال الاول :

في هذا المثال سوف نتعرف على قواعد البيانات الموجودة لدينا .

test.php - المفكرة

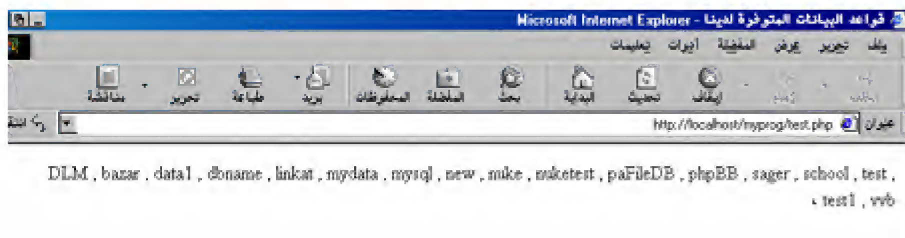
ملف تحرير بحث تعليمات

```
<HTML dir=rtl>
<HEAD>
<TITLE> قواعد البيانات المتوفرة لدينا </TITLE>
</HEAD>
<BODY>
<?
// الاتصال مع سيرفر قواعد البيانات
$Link = mysql_connect("localhost", "root", "");

// الحصول على قواعد البيانات المتوفرة
$dbResult = mysql_list_dbs($Link);

// طباعة قواعد البيانات
while($row = mysql_fetch_row($dbResult))
        {
                print($row[0] . " , ");
        }
?>
</BODY>
</HTML>
```

وعند تنفيذ هذا المثال نحصل على النتيجة التالية :



المثال الثاني :

فـــي هـــذا المـــثال ســـوف نـــتعرف على الجداول الموجودة في قاعدة البيانات المسماة **school** .

المفكرة - test.php

ملف تحرير بحث تعليمات

```
<HTML dir = rtl>
<HEAD>
<TITLE> الـجـد اول فـي قـا عـد ة الـبـيـانـات </TITLE>
</HEAD>
<BODY>
<?
// الاتـصـال مـع سـيـر فـر قـو ا عـد الـبـيـانـات
$Link = mysql_connect("localhost", "root", "");

// الـحـصـول عـلـى الـجـد اول
$dbResult = mysql_list_tables("school");

// طـبـاعـة الـجـد اول
while($row = mysql_fetch_row($dbResult))
        {
                print($row[0] . "<BR>\n");
        }
?>
</BODY>
</HTML>
```

عند تنفيذ هذا المثال سوف نحصل على جدول واحد هو **std** .

تســـتطيع الحصـــول علـــى الجداول في قاعدة بيانات اخرى عن طريق تغيير اسم قاعدة البيانات فقط .

المثال الثالث :

فـــي هـــذا المثال سوف نتعرف على الحقول المتوفرة في الجدول **std** ومواصفات هذه الحقول .

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<HTML>
<HEAD>
<TITLE> الحقول المتوفرة في الجدول ومعلوماتها </TITLE>
</HEAD>
<BODY>
<?
// الاتصال مع سيرفر قواعد البيانات
$Link = mysql_connect("localhost", "root", "");
// الحصول على الحقول الموجودة في الجدول
$dbResult = mysql_list_fields("school", "std");
// تجهيز المخرجات على شكل جدول
        print("<TABLE>\n");
        print("<TR>\n");
        print("<TH>Name</TH>\n");
        print("<TH>Type</TH>\n");
        print("<TH>Length</TH>\n");
        print("<TH>Flags</TH>\n");
        print("</TR>\n");
// طباعة المخرجات
for($i = 0; $i < mysql_num_fields($dbResult); $i++)
        {
print("<TR>\n");
print("<TD>".mysql_field_name($dbResult,$i)."</TD>\n");
print("<TD>".mysql_field_type($dbResult,$i)."</TD>\n");
print("<TD>".mysql_field_len($dbResult,$i)."</TD>\n");
print("<TD>".mysql_field_flags($dbResult,$i)."</TD>\n");
print("</TR>\n");
        }
print("</TABLE>\n");
?>
</BODY>
</HTML>
```

عند تنفيذ هذا المثال نحصل على التالي :

Name	Type	Length	Flags
no	int	11	not_null
name	string	10	not_null
lastname	string	10	not_null
gride	int	11	not_null

المثال الرابع :

فـــي هـــذا المثال سوف نتعرف على كيفية اجراء استعلام من قاعدة البيانات واستخراج البيانات ووضعها في تنسيق جميل .

المفكرة - test.php

ملف تحرير بحث تعليمات

```
<?php
echo "<html dir=rtl>";
echo "<TITLE> استعراض السجلات </TITLE>" ;
$link = mysql_connect ("localhost", "root","");
$result = mysql_db_query ("school","select * from std");

echo "<table border=1 cellpadding=0 cellspacing=0
style=border-collapse: collapse bordercolor=#111111
width=50%>" ;
echo "<td width=10%
align=center bgcolor=#FF0000> الرقم </td>";
echo "<td width=30%
align=center bgcolor=#FF0000> الاسم الاول </td>";
echo "<td width=30%
align=center bgcolor=#FF0000> الاسم الثاني </td>";
echo "<td width=30%
align=center bgcolor=#FF0000> الدرجة </td>";

while ($row = mysql_fetch_array ($result)) {
echo  "<tr>";
echo "<td width=10%
align=center bgcolor=#FF00FF> $row[no] </td>";
echo "<td width=30% align=center> $row[name] </td>";
echo "<td width=30% align=center> $row[lastname] </td>";
echo "<td width=30% align=center> $row[gride] </td>";
echo  "</tr>" ;
}
echo  "</table>" ;
mysql_free_result ($result);
mysql_close ($link);
?>
```

عند تنفيذ هذا المثال سوف نشاهد التالي :



الرقم	الاسم الاول	الاسم الثاني	الدرجة
١	sager	mohammed	٩٥
٢	khaled	ahmad	٩٣
٣	sager	sultan	٨٦
٥	sager	abrahem	٦٧
٦	yosef	rakan	٩٢

امثلة متقدمة في PHP

استخدام الــ Cookies

في بعض الاحيان تود ان تعرف هل زار هذا الشخص صفحتك من قبل او تود ان لايدخل كلمـــة الســـر فـــي زيارته التالية بل تتعرف عليها عن طريق ملفات صغيرة تضعها في جهازه وهذه الملفات تدعى **Cookies** .

اذا كنــت من رواد المنتديات فسوف تلاحظ انه عند زيارتك للمنتدى مباشرة تجد رسالة ترحيــب بك وايضا يمكنك اضافة مشاركة بدون الحاجة لكتابة اسمك وكلمة السر وهذه وظيفة الــكوكيز التي قام من قبل بزراعتها في جهازك .

اذن دعنا نتعلم كيف نقوم بزراعة الكوكيز في اجهزة زوار موقعك والاستفاده منها .

سوف ندرس هنا مثال مفصل للكوكيز ونحتاج الى ان ننشئ ملفين لهذا المشروع .

الملــف الاول اســمه **prog1.php** وسوف يقوم بأستقبال الزائر وتسجيل اسمه في ملف كوكيز .

الملــف الثانــي اســمه **prog2.php** وســوف يقوم بعرض الاسم المخزن بملف الكوكيز .

الملف الاول (**prog1**) ونضع فيه الاوامر التالية :

المفكرة - prog1.php

ملف تحرير بحث تعليمات

```
<?php

if ($name) {
$expires = time()+(3600);
setcookie ("myname" , $name ,"$expires");
echo "<html dir=rtl>" ;
print "اهلا بك يا   $name   " ;
echo '<a href="prog2.php"> ادخل الى الموقع   </a>';
echo '<p><a href="prog1.php"> لتغيير الاسم اضغط هنا
</a></p>';

} else {

echo "<html dir=rtl>" ;
echo "<form method=POST action=prog1.php>" ;
echo "<span lang=ar-sa>ادخل اسمك هنا </span>" ;
echo "<input type=text name=name size=20><input
type=submit value=احفظ name=B1></p>" ;
echo "</form>" ;
}

?>
```

عند تنفيذ هذا المثال سوف نشاهد التالي :



كمـــا تلاحـــظ مربع نص لادخال الاسم وعند ادخال الاسم والضغط على الزر احفظ ننتقل الى الشاشة التالية :



وعــند الضغط على ادخل الى الموقع سوف ينقلنا الى البرنامج الثاني الذي سوف نكتبه بعد قليل .

اما عند الضغط على لتغيير الاسم فسوف يعيدنا الى الصفحة السابقة .

وجميع هاتين الشاشتين ينتجهما البرنامج الذي كتبناه وهو (**prog1**) .

ولكــن هل تعلم الان انك زرعت ملف صغير في جهازك به هذه المعلومات وهو موجود في دليل وندوز في المجلد **Cookies** .

ولزراعة أي ملف كوكيز كل ماعليك هو كتابة الامر التالي :

setcookie ("اسم المتغير" ، "قيمته" ، "تاريخ نهاية الكوكيز ")؛

مثال :

```
setcookie ("myname" , “sager” ," time()+(3600)");
```

الان لنعود لمثالنا ولنعمل الملف الثاني .

الملف الثاني (**prog2**) ونضع فيه الاوامر التالية :

prog2.php - المفكرة

ملف تحرير بحث تعليمات

```
<?

if ($myname) {
echo "<html dir=rtl>" ;
print "اهلا بك يا   $myname " ;

} else {
print "لم تقم بالتسجيل " ;
echo '<p><a href="prog1.php"> اضغط هنا للتسجيل </a></p>';

}

?>
```

وهــذا الملف عند الدخول الية يتأكد هل ملف الكوكيز موجود في جهاز الزائر ، فاذا كان موجود قرأ الاسم وعرض الرسالة الترحيبية واذا كان غير موجود اظهر رسالة تبين انه لم يقم بالتسجيل ويضع له وصلة الى البرنامج الاول .

ملاحظة : يجب ان يكون الملفين في مجلد واحد .

برنامج لإضافة واستعراض النكت

سـنقوم بكـتابة بـرنامج يتيح للمستخدم كتابة نكت وحفظها في قاعدة البيانات وايضا يستطيع الزائر استعراض النكت الموجودة كل هذا في ملف واحد .

في البداية نحتاج الى انشاء قاعدة بيانات جديدة ونسميها (**jokes**) .

ننشىء في قاعدة البيانات جدول بنفس الاسم (**jokes**) .

في الجدول ننشىء حقلين كما يلي :

الاول اسمه (**JokeText**) ونوعه (**text**) .

الثاني اسمه (**JokeDate**) ونوعه (**date**) .

الان نفـتح المذكـرة ونسـمى الملف بأي اسم ينتهي بالامتداد **php** ونكتب به الاوامر التالية :

test.php - المفكرة

ملف تحرير بحث تعليمات

```
<html dir=rtl>
<?php
if (isset($addjoke)):?>

<FORM ACTION="<?php echo($PHP_SELF); ?>"
METHOD=POST><P>اكتب النكته التي تريد اضافتها<BR>
<TEXTAREA NAME="joketext" ROWS=10 COLS=40 WRAP>
</TEXTAREA><BR><INPUT TYPE=SUBMIT NAME="submitjoke"
VALUE="اضف"></FORM>

<?php  else:
$dbcnx = @mysql_connect("localhost","root", "");
if (!$dbcnx) {
echo ( "<P> لم استطع الاتصال بالسيرفر </P>" );
exit();
   }

if (! @mysql_select_db("jokes") )
{
echo ( "<P> لم استطع الاتصال بقاعدة البيانات </P>" );
exit();
}
```

```
if ("اضف" == $submitjoke) {
$sql = "INSERT INTO Jokes SET JokeText='$joketext',
 JokeDate=CURDATE()";

if (mysql_query($sql)) {
echo ("<P> لقد تم اضافة النكتـه التـي كتبتـهـا </P>");
} else {
echo("<P>حدث الخطأ التـالـي : " . mysql_error() . "</P>");

}
}
echo("<P> هذه جميع النـكت الموجودة في قاعدة البـيـانـات </P>");

$result = mysql_query("SELECT JokeText FROM Jokes");

if (!$result) {
echo ("<P>حدث الخطأ التـالـي :  " . mysql_error() . "</P>");

exit();
}
while ( $row = mysql_fetch_array($result) ) {
echo("<P>" . $row["JokeText"] . "</P>");
}
echo("<P><A HREF='$PHP_SELF?addjoke=1'>"."اضف نـكتـه</A></P>");
endif;
?>
```

الان عند كتابة هذه الاوامر وتنفيذ البرنامج نشاهد الشاشة التالية :



هذه جميع النكت الموجودة في قاعدة البيانات

اضف نكته

وحيــث ان هذه اول مرة ننفذ فيها البرنامج نجد انه لاتوجد أي نكت ، الان نضغط على الرابط اضف نكته لنتجه الى صندوق اضافة النكت كما يلي :



اكتب النكته التي تريد اضافتها

برنامج نكووووووت النكته الرابعة

اضف

اكتـب النكته التي تريد اضافتها ثم اضغط الزر اضف ، كرر العملية لترى الشـكل التالي :



لقد تم اضافة النكته التي كتبتها

هذه جميع النكت الموجودة في قاعدة البيانات

النكته الاولى

النكته الثانية

النكته الثالثة

برنامج نكووووووت النكته الرابعة

اضف نكته

مسجل معلومات الزوار

الان سوف نقوم بكتابة برنامج يسجل معلومات زوار موقعك في ملف داخل موقعك تستطيع الاطلاع عليه وقراءته .

هذا البرنامج يتكون من جزئين :
الملف الاول :وسوف نسميه (first.php) سوف يسجل معلومات الزائر وايضا يعرض المعلومات على الزائر ، والمعلومات التي تسجل هي رقم الـ ip ومنفذ الاتصال ونوع المتصفح .
الملف الثاني : وسوف نسميه (show.php) هذا الملف فقط للمسئول وعن طريقه تستعرض جميع معلومات زوار موقعك .

نكتب الاوامر التالية في الملف الاول (first.php) :

first.php - المفكرة

ملف تحرير بحث تعليمات

```
<html dir="rtl">

<?php
$ip = getenv ("REMOTE_ADDR");
$port = getenv ("REMOTE_PORT");
$browser=getenv ("HTTP_USER_AGENT");

$v_fopen = fopen ("inf.txt","a+");
fwrite ($v_fopen ,"$ip *** $port *** $browser\n");
fclose($v_fopen);

print "<p> اهلا بك عزيزي الزائر هذه معلوماتك </p>" ;
print "<table border=1 cellpadding=0
cellspacing=0 style=border-collapse: collapse
bordercolor=#111111 width=50%> " ;
print  "<tr>" ;
print  "<td width=1% align=center>ip</td>" ;
print  "<td width=16% align=center>المنفذ</td>" ;
print  "<td width=64% align=center>المتصفح</td>" ;
print  "</tr>" ;
print  "<tr> " ;
print  "<td width=20% align=center>$ip</td>" ;
print  "<td width=16% align=center>$port</td>" ;
print  "<td width=64% align=center>$browser</td>" ;
print  "</tr>" ;
print  "</table>" ;

?>
```

وايضا نكتب الاوامر التالية في الملف الثاني (show.php) :

المفكرة - show.php

ملف تحرير بحث تعليمات

```
<html dir="rtl">

<?php

$readfile = file ("inf.txt");

print "<p> اهلا بك هذه معلومات جميع زوار موقعك </p>" ;
print "<table border=1 cellpadding=0
cellspacing=0 style=border-collapse: collapse
bordercolor=#111111 width=50%> " ;
print  "<tr>" ;
print  "<td width=1% align=center>ip رقم </td>" ;
print  "<td width=16% align=center>المنفذ </td>" ;
print  "<td width=64% align=center>المتصفح </td>" ;
print  "</tr>" ;

while (list (,$eachline) = each ($readfile)) {

$line = explode(" *** ",$eachline) ;

print  "<tr> " ;
print  "<td width=20% align=center>$line[0]</td>" ;
print  "<td width=16% align=center>$line[1]</td>" ;
print  "<td width=64% align=center>$line[2]</td>" ;
print  "</tr>" ;

}

print  "</table>" ;

?>
```

وبعد كتابة الكود وحفظ الملفين ننفذ الملف الاول ، وسوف نشاهد رسالة ترحب بالزائر بالاضافة الى معلومات هذا الزائر انظر الشكل :



اهلا بك عزيزي الزائر هذه معلوماتك

ip	المنفذ	المتصفح
١٢٧.٠.٠.١	١٠٦٠	Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)

ولكي تطلع على معلومات جميع زوار موقعك نفذ الملف الثاني لتشاهد الشكل التالي :



اهلا بك هذه معلومات جميع زوار موقعك

رقم ip	المنفذ	المتصفح
١٢٧.٠.٠.١	١٠٥٨	Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
١٢٧.٠.٠.١	١٠٥٩	Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
١٢٧.٠.٠.١	١٠٦٠	Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)

مستعرض الصور الالي

الان سـوف نكتب برنامج يبين قوة PHP في التعامل مع السيرفر بحيث يقوم الـبرنامج بقـراءة الصـور الموجـودة فـي نفس المجلد وعرضها بشكل آلي ، أي صورة خلف صورة .

نحتاج فقط لملف واحدة سوف نسميه (pic.php)
وسوف نكتب فيه الاوامر التالية :

pic.php - المفكرة

ملف تحرير بحث تعليمات

```
<html dir=rtl>
<head><title> مستعرض الصور الآلي </title>
<?
$handle=opendir('.');
$counter = 0;
while ($file = readdir($handle)) {
   $the_type = strrchr($file, ".");
   $is_picture = eregi("jpg|gif|bmp|png",$the_type);
   if ($file != "." and $file != ".." and $is_picture) {
      $mypics[$counter] = $file;
      $counter++;
      }
   }
closedir($handle);
?>
<script language="Javascript">
<!--
<?
$firstpic = $mypics[0];
print('  var imagearray = new Array(');
for ($i=0; $i < ($counter-1); $i++) { print("'" . $mypics[$i] .
"', "); }
print("'" . $mypics[$counter-1] . "');\n");
?>
   mystring = new Array;
   maximum = imagearray.length;
   for (i = 0; i < maximum ; i++) {
      mystring[i] = new Image();
      mystring[i].src = imagearray[i];
   }
```

```
var rotation = 0;
var mydelay = 2000;
function rotate(){
  if (document.images) { document.images.rotate.src =
mystring[rotation].src; }
  else { document.all.rotate.src = mystring[rotation].src; }
  rotation++;
  if (rotation == maximum) {
  rotation = 0;
  mytimer = setTimeout("rotate()", (mydelay * 2));
    }
  else { mytimer = setTimeout("rotate()", mydelay); }
}
//-->
</script>
</head>
<body bgcolor="#ffffff" onload="rotate();">
<center><h2>انتظر سوف يستمر العرض</h2>
<img src="<? print($firstpic); ?>" name="rotate">

<p>
<a href="javascript:history.back()"> للعودة اضغط هنا </a>
</center>
</body>
</html>
```

بعد كتابة البرنامج وحفظه ، نضع معه في نفس المجلد عدة صور ، ثم ننفذ البرنامج لنحصل على التالي :



عداد الزوار

سـوف نكتب برنامج متكامل يقوم بعد الزوار الذين يزورون صفحتك ، بل سوف نجعلـه يخـرج النـتائج علـى شكل صور ، وايضا سوف يكون بسيطا جدا ليمكن استيعابه بكل سهولة .

نحتاج الى مايلي :
ملف واحد سوف نكتب به الاوامر وسوف نسميه (count.php) .
مجلد اسمه (pic) يحتوي على الصور من صفر الى تسعه .
ملف فارغ اسمه (data.txt) سوف نضع فيه عدد الزوار اثناء التنفيذ .

الان نكتب الاوامر في الملف (count.php) كما يلي :

count.php - المفكرة

ملف تحرير بحث تعليمات

```
<?php
// الملف الذي تخزن به اعداد الزوار
$COUNT_FILE = "data.txt";
// اين توجد صور الارقام
$IMG_DIR_URL = "./pic/";
// عدد الخانات التي سوف تطبع
$NB_DIGITS = 8;

if (file_exists($COUNT_FILE)) {
$fp = fopen("$COUNT_FILE", "r+");
flock($fp, 1);
$count = fgets($fp, 4096);
$count += 1;
fseek($fp,0);
fputs($fp, $count);
flock($fp, 3);
fclose($fp);
} else {
echo "خطأ : لم اجد ملف تخزين البيانات";
exit;
}
```

```
chop($count);
$nb_digits = max(strlen($count), $NB_DIGITS);
$count = substr("0000000000".$count, -$nb_digits);
$digits = preg_split("//", $count);
for($i = 0; $i <= $nb_digits; $i++) {

if ($digits[$i] != "") {
$html_result .=  "<IMG
SRC=\"$IMG_DIR_URL$digits[$i].gif\">";
}

}
?>

<?php echo $html_result; ?>
```

الان احفظ الملف ونفذ الملف عن طريق المتصفح لتشاهد التالي :



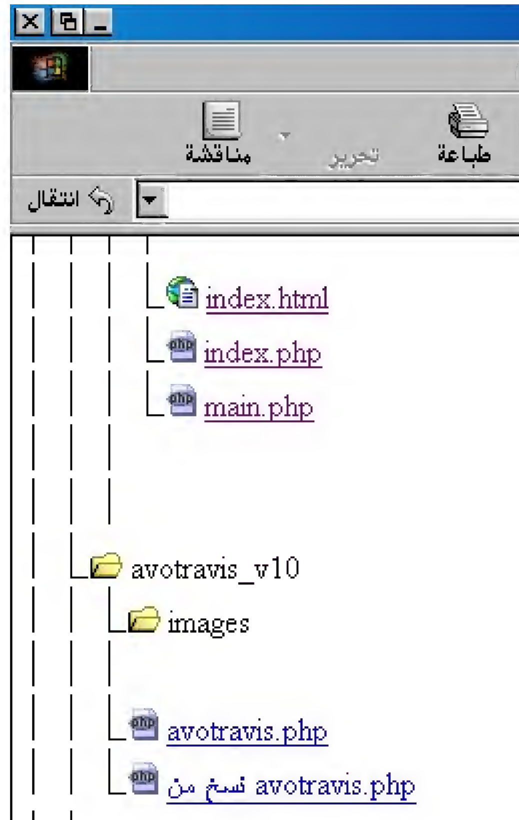
بالطبع نستطيع اضافة هذا الملف في أي صفحة او ملف .

خريطة الموقع

هـذا البرنامج يعرض جميع ملفات (php & html) اعتبارا من المجلد الحالي الذي نضع فيه هذا البرنامج مما يتيح للزائر الانتقال الى أي موقع مباشرة .
البرنامج يتكون من ملف واحد ولكن لن نكتب السورس كوود له هنا لانه طويل ولكـن سوف تجده في القرص المدمج المرفق ولكن سوف نشرح طريقة عمل البرنامج والنتيجة التي تظهر عند تنفيذ البرنامج .

نحتاج الى مايلي :
ملف البرنامج اسمه (sitemap.php) موجود به جميع الاوامر .
٦ ملفات صور لعرض اشكال الملفات والمجلدات والخطوط التي نحتاجها .

الان انقل الملف من القرص المدمج الى الموقع الرئيسي لموقعك ونفذ الملف لتشاهد النتيجة التالية :



لعبة (Tic-Tac-Toe) الشهيرة

هل تعتقد انك لاتستطيع ان تصنع العابا بواسطة لغة php ؟ اذن انت مخطئ .
الان سوف نتعرف على كيفية كتابة لعبة Tic-Tac-Toe الشهيرة وكما فعلنا سابقا بكل سهولة وبواسطة ملف واحد فقط .

نحتاج الى مايلي :
ملف واحد اسمه (tic.php) لنكتب فيه جميع الاوامر .
٤ صور تعبر عن علامة الضرب والدائرة و الخطوط وصورة خالية .

الان نكتب في الملف (tic.php) الاوامر التالية :

tic.php - المفكرة

ملف تحرير بحث تعليمات

```
<?php
  if ($game==0)
  {
    $a=0; $b=0; $c=0;
    $d=0; $e=0; $f=0;
    $g=0; $h=0; $i=0;
    $player=1;
  }
  else
  {
    $player = (++$player > 2) ? 1 : 2;
  }
  $pieces = array ( $a,$b,$c,$d,$e,$f,$g,$h,$i );
?>

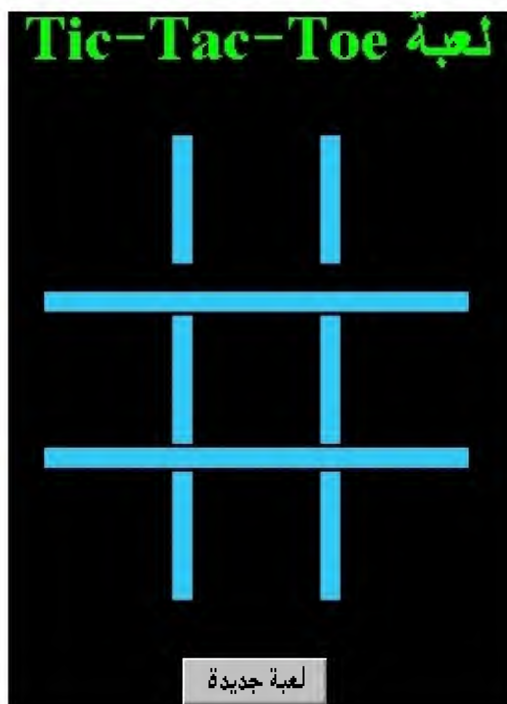
<?php
   function draw_piece($idx,$p,$player)
   {
     switch ($p[$idx])
     {
  case 0:  //فارغ
   $p[$idx]=$player;
   $url = "<a href=\"tic.php?game=1&";
   $url .= "a=$p[0]&b=$p[1]&c=$p[2]&";
   $url .= "d=$p[3]&e=$p[4]&f=$p[5]&";
   $url .= "g=$p[6]&h=$p[7]&i=$p[8]&";
   $url .= "player=$player";
   $url .= "\">";
   $url .= "<img src=\"blank.gif\"
   width=64 height=64 border=0></a>";
   print $url;
   break;
```

```
   case 1: // الدائرة
    print "<img src=\"ro.gif\" border=0>";
    break;
   case 2: // علامة الضرب
    print "<img src=\"bx.gif\" border=0>";
    break;
   default: // افتراضي فارغ
    print "<img src=\"blank.gif\" border=0>";
    break;
      }
    }
?>

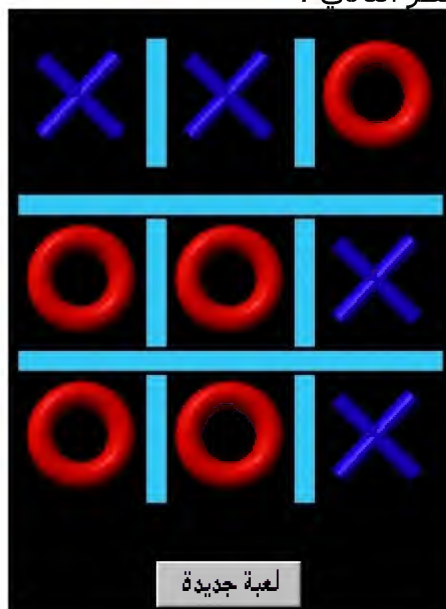
<html>
  <body bgcolor="#000000">
  <font color="#00FF00">
    <center><h1> Tic-Tac-Toe لعبة</h1></center>
  </font>
<table align=center cellpadding=0 cellspacing=0 border=0>
    <tr>
      <td>
        <? draw_piece(0,$pieces,$player) ?>
      </td>
      <td><img src="pix.gif" width=10 height=64></td>
      <td>
        <? draw_piece(1,$pieces,$player) ?>
      </td>
      <td><img src="pix.gif" width=10 height=64></td>
      <td>
        <? draw_piece(2,$pieces,$player) ?>
      </td>
      </tr>
    <tr bgcolor=blue height=10></tr>
    <tr>
```

```
      <td><img src="pix.gif" width=64 height=10></td>
      <td><img src="pix.gif" width=10 height=10></td>
      <td><img src="pix.gif" width=64 height=10></td>
      <td><img src="pix.gif" width=10 height=10></td>
      <td><img src="pix.gif" width=64 height=10></td>
    </tr>
    <tr>
      <td>
        <? draw_piece(3,$pieces,$player) ?>
      </td>
      <td><img src="pix.gif" width=10 height=64></td>
      <td>
        <? draw_piece(4,$pieces,$player) ?>
      </td>
      <td><img src="pix.gif" width=10 height=64></td>
      <td>
        <? draw_piece(5,$pieces,$player) ?>
      </td>
    </tr>
    <tr>
      <td><img src="pix.gif" width=64 height=10></td>
      <td><img src="pix.gif" width=10 height=10></td>
      <td><img src="pix.gif" width=64 height=10></td>
      <td><img src="pix.gif" width=10 height=10></td>
      <td><img src="pix.gif" width=64 height=10></td>
    </tr>
    <tr>
      <td>
        <? draw_piece(6,$pieces,$player) ?>
      </td>
      <td><img src="pix.gif" width=10 height=64></td>
      <td>
        <? draw_piece(7,$pieces,$player) ?>
      </td>
      <td><img src="pix.gif" width=10 height=64></td>
      <td>
        <? draw_piece(8,$pieces,$player) ?>
      </td>
    </tr>
</table>
<br>
  <FORM ACTION='tic.php?game=0' METHOD=POST>
    <center>
      <Input Type=submit Value='لعبة جديدة'>
    </center>
</body>
</html>
```

الان احفظ الملف ، ومن خلال المتصفح نفذ الملف لتشاهد التالي :



الان استمتع باللعبة انظر التالي :



الوقت والتاريخ بالصور

الان سـوف نـتعرف عـلـى كيفـية الحصـول عـلـى الوقـت والتاريخ بشكل جميل وبكل سـهولة .

نحتاج الى مايلي :
ملف واحد اسـمـه (clock.php) موجود به جميع الاوامر .
١٤ صورة تعبر عن الارقام والفواصل .

الان نكتب في الملف (clock.php) الاوامر التالية :

clock.php - المفكرة

ملف تحرير بحث تعليمات

```
<?php
$IMG_DIR_URL = "./digits/";

function txt2digits ($txt) {
global $IMG_DIR_URL;
$digits = preg_split("//", $txt);
while (list($key, $val) = each($digits)) {
if ($val != "")  {
if ($val == "-") $val = "dash";
if ($val == ":") $val = "colon";
$html_result .= "<IMG SRC=\"$IMG_DIR_URL$val.gif\">";
}
}
return $html_result;
}

$date = txt2digits(date("d-m-Y"));
$time = txt2digits(date("H:i:s"));
?>
<BODY BGCOLOR="white">
<BR><BR><P ALIGN="Center">الساعة و التاريخ </P><BR>
<CENTER>
<?php echo $date ?>
<P>
<?php echo $time ?>
</CENTER>
```

وعند تنفيذ الملف نشاهد التالي :

الساعة والتاريخ

06-11-2000

15:35:51

بالطبع تستطيع اضافة الملف في أي صفحة .

تركيب المجلة Php Nuke

الكثير من المواقع بدأت في تركيب المجلة الالكترونية Php Nuke بل اصبحت هذه المجلة هي المدير الاساسي للموقع لسببين مهمين :
الاول : ان هذه المجلة هي الافضل حتى الان .
الثاني : ان هذه المجلة مجانية .
تتكون المجلة من الكثير من البرامج المجتمعة منها :
برنامج لادارة الاخبار ويعتبر حتى هذه اللحظة الأفضل .
برنامج لادارة تنزيل البرامج .
برنامج لادارة روابط المواقع .
برنامج لادارة الاعلانات .
برنامج لادارة الاستفتاءات .
والكثير من البرامج الخدمية الاخرى .

انظر شكل المجلة :



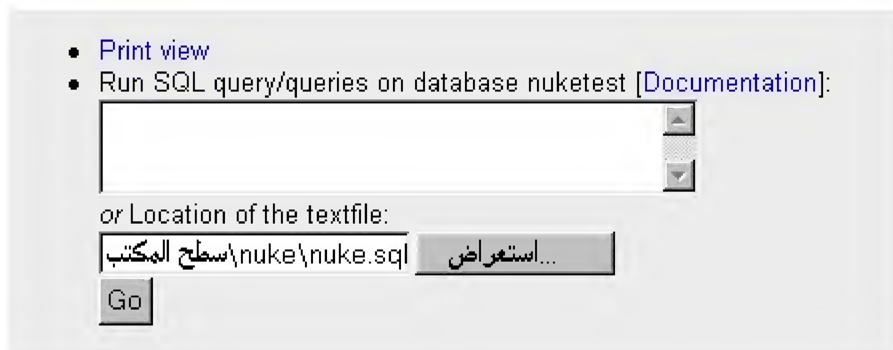
هذه المجلة تدل على قوة لغة php وامكانية تطويعها بشكل احترافي ، لقد قمنا بتعريب هذه المجلة ووضعنا التعريب في السي دي المرفق مع الكتاب ، وسوف نقوم هنا بشرح خطوات تركيب وادارة المجلة .

خطوات تركيب المجلة

- في البداية انقل المجلد (nuke) وجميع الملفات الموجودة فيه الى موقعك او الى السيرفر الشخصي الموجود في جهازك .
- الان افتح الملف (config.php) الموجود في المجلد (nuke) وغير البيانات الضرورية كما في الصورة :

```
$dbhost = "localhost";   //  اسم السيرفر غالبا لن تحتاج الى تغييره
$dbuname = "root";       //  اسم المستخدم لقاعدة البيانات
$dbpass = "";            //  كلمة السر لقاعدة البيانات
$dbname = "nuketest";    //  اسم قاعدة البيانات
$prefix = "nuke";        //  لاتغير هذا السطر
$user_prefix = "nuke";   //  لاتغير هذا السطر
$dbtype = "MySQL";       //  لاتغير هذا السطر
```

- الان عن طريق المتصفح اتجه الى البرنامج (PhpMyAdmin) لتحكم بقواعد البيانات واضغط في اليسار على القاعدة التي حددتها للبرنامج .
- الان اضغط على الزر استعراض واختر الملف (nuke.sql) انظر الشكل :



- الان اضغط الزر (Go) ليبدأ البرنامج بأنشاء الجداول والحقول في قاعدة البيانات .
- الان اصبحت المجلة جاهزة للعمل كل ماعليك فعله هو كتابة العنوان الصحيح في المتصفح .

ادارة المجلة

للدخول الى برنامج ادارة المجلة نحتاج الى كتابة العنوان التالي في المتصفح او مايشابهه لديك (http://localhost/nuke1/admin.php) وعند ذلك نشاهد نافذة تطالب بالاسم وكلمة السر كما يلي :

تسجيل الدخول للتحكم
رقم المشرف sager
كلمة المرور ******
دخول

اذا كنت ركبت النسخة المعربة الموجودة مع هذا الكتاب يكون الاسم وكلمة السر كما يلي :

رقم المشرف : sager
كلمة السر : ١٢٣٤٥٦

اما اذا كانت نسخة اخرى فغالبا يكون الاسم وكلمة السر كما يلي :

رقم المشرف : good
كلمة السر : password

الان اضغط الزر دخول للولوج الى لوحة التحكم لتشاهد الشاشة التالية :



سنتعرف الان على عمل اهم هذه الايقونات كما يلي :



هذه الايقونة تساعدنا في اضافة مقال او خبر جديد وعند الضغط على هذه الايقونة سوف ننتقل الى شاشة ادخال الخبر كما يلي :

العنوان : هو العنوان الذي يظهر في رأس الخبر .
الموضوع : عليك ان تحدد المجموعة التي ينتمي هذا الخبر اليها .
هل انشرها في الصفحة الاولى : عند اختيار نعم سوف يوضع هذا الخبر في الصفحة الاولى .
نص المقال : هنا تضع ملخص المقال .
النص الاضافي : بقية المقال عندما يضغط الزائر كلمة المزيد .
في الاسفل سوف تجد احد خيارين (استعراض المقال ، ارسل المقال) ويعبران عن مايلي :
استعراض المقال : لاستعراض المقال قبل حفظه .
ارسل المقال : لحفظ المقال في القاعدة وعرضه للزوار .



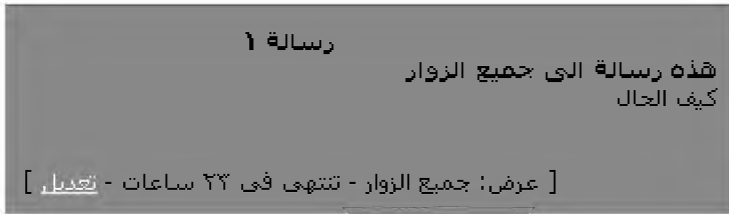
عـند الضـغط علـى هـذه الايقونـة نسـتطيع اضافة او تعديل بيانات عضو ، وعند الضغط على هذه الايقونة يضهر امامنا قسـمين :

القسـم الاول : لتعديل او حذف عضو ، وكل ماعليك فعله هو كتابة كنية العضو ثم اختيار العملية المراد تنفيذها ثم الضغط على الزر موافق .

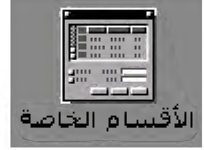
القســم الثانـي : اضـافة عضـو جديـد ، وهنا يتم تعبئة بيانات العضو الجديد مع العلم ان البيانات الموجودة امامهما كلمة مطلوب يجب كتابتها .



هـذه الايقونة مخصصة لوضع رسـائل تظهر في الصفحة الاولى كخبر ليسـتطيع الزائر المحدد من قراءتها .
وهذا نموذج لرسـالة :



وعند الضغط على هذه ايقونة الرسـائل تظهر البيانات التالية :
العنوان : عنوان الرسـالة ويظهر في اعلى الرسـالة
المحتوى : يعبر عن نص الرسـالة .
تاريخ الانتهاء : الفترة التي ينتهي عرض الرسـالة بعدها .
Active : هل الرسـالة نشـطة في هذه اللحظة ام لا .
من يسـتطيع مشـاهدة هذا : تحديد الزوار الذين يحق لهم مشـاهدة الرسـالة .
الزر (Add message) : حفظ الرسـالة وعرضها .



هـذه الايقونـة تنشـئ اقسـام خاصة قد لا يكون لها علاقة بالاقسام الرئيسية فـي الموقـع ويـتم الوصـول الـيها عن طريق القائمة في الصفحة الرئيسية من الرابط الاقسام .
عند الضغط على هذه الايقونة نجد امامنا الاقسام التالية :

القسم الاول : الاقسام النشطة الحالية وعند الضغط على أي قسم نستطيع تعديل اسمه او ايقونته او مسحه تماما .

القسـم الثاني : اضف مقالا جديد في القسـم ، هنا نستطيع اضافة المقالات الجديدة وتحديد المقال الذي يتبع له القسم .

القسم الثالث : نجد فيه آخر ٢٠ مقال تم كتابتهم .

القسم الرابع : اضف قسم جديد وهنا نضع اسم القسم والصورة التي نريد ان تظهر معه ليصبح لدينا قسم خاص جديد .



عـند طـريق هـذا القسـم نسـتطيع الحصـول على نسخة احتياطية من قاعدة البـيانات ، فمـثلا عـندما نجـري الكثير من التعديلات في البرنامج في سيرفرنا الشخصـي ونـود في النهاية نقل هذه التعديلات الى موقعنا فاننا نأخذ نسخة من هذه البيانات عن طريق هذا القسم .
عند الضغط على هذه الايقونة سوف تظهر شاشة تطالبك بتحديد الاسم لهذا الملـف مـع انـه يحـدد له اسـم افتراضـي تاريخ اليوم ، الان حدد الموقع المراد حفظ النسخة فيه ثم اضغط موافق .



عند الضغط على هذه الايقونة نستطيع التحكم بجميع البريمجات الموجودة لدينا عن طريق تنشيطها او ايقافها عن العمل او تغيير عنوانها .



عـندما يرسـل الاعضـاء مواضـيع او اخبار جديدة اليك لنشرها في موقعك فانها سـوف تكـون موجـودة فـي هـذا القسـم قـبل نشـرها لـتطلع علـيها وتعطـي موافقتك على نشرها او حذفها دون نشرها .



المجلـة تـتكون مـن مجموعـات مخـتلفة يـتم تركيـبها بشـكل جمـيل حسـب الترتيـب الـذي نـراه مناسـب ، كمـا انـه توجـد الكثير من المجموعات الجاهزة التـي يـتم تحديـد المجموعـة التـي تظهـر للـزوار المختلفين ايضا يمكن اضافة مجموعـات خارجـية للمجلـة مـثل مجموعـة تعـرض بشـكل عشـوائي آيـة من القرآن الكريم او مجموعة تنشر خبر من احد مصادر الاخبار وهكذا .



هل تود معرفة آراء زوارك في موضوع معين ؟
اذن هذا القسـم الذي تبحث عنه ، فهذا القسـم يقوم بانشاء استفتاءات او تصويتات عديدة .



عندما يكون لديك منتج فستجد من الزبائن اسئلة متكررة كيف اعمل هذا ؟ او لماذا لم يحدث هذا ؟ ولكي توفر الوقت تكتب كل سؤال جديد هنا وتكتب الاجابة عليه ومن ثم على الزبون قراءة هذه الاسئلة قبل الاتصال بك فاذا لم يجد سؤاله اتصل بك .



الاخبار والمواضيع التي نكتبها في المجلة تكون تحت عناوين او اقسام عامة مثل قسم البرمجة و قسم التصميم وغيرها وعند طريق هذه الايقونة تستطيع انشاء الاقسام التي تريد في مجلتك .



في هذا القسم تستطيع ان تنشيء مكتبة تحتوى على برامج متاحة لزائر موقعك لتنزيلها ووضع هذه البرامج على شكل اقسام عامة وفرعية .



عن طريق هذا القسم تستطيع تغيير المعلومات العامة لموقعك والمعلومات التي تظهر في اسفل المجلة ، وكذلك الصور وايضا تغيير الشكل العام للمجلة فلو كان لديك شكل آخر يمكن من هنا ان تضيفه .



بواسـطة هـذا القسـم نسـتطيع ان نصـنع مكتـبة للمواقع مقسمه الى مواقع اسلامية ومواقع برمجة وغيرها حسب التقسييم الذي نريد .

عـند الدخـول الـى هـذا القسـم نسـتطيع التحكم بجميع الروابط وانشاء روابط جديدة ومعرفة الروابط التي لاتعمل .



بواسطة هذا القسم نستطيع اعطاء صلاحيات المشرفين على المجلة وتستطيع ايضا تغيير كلمة السر الخاصة بك .



عن طريق هذا القسم تستطيع معرفة العناوين التي تم الوصول منها او الاتصال بها من خلال المجلة ، عند عدم الحاجة لهذه العناوين احذفها .